Network-provider-independent overlays for resilience and quality of service.

Zhang, Xian

For additional information about this publication click this link.
http://qmro.qmul.ac.uk/jspui/handle/123456789/2443

Network-provider-independent Overlays for Resilience and Quality of Service

by

Xian ZHANG

A thesis submitted to the University of London for the degree of
the Doctor of Philosophy

School of Electronic Engineering and Computer Science
Queen Mary, University of London
United Kingdom

September 2011

To my family and those who inspire, care and support me through the student life of mine

Abstract

Overlay networks are viewed as one of the solutions addressing the inefficiency and slow evolution of the Internet and have been the subject of significant research. Most existing overlays providing resilience and/or Quality of Service (QoS) need cooperation among different network providers, but an inter-trust issue arises and cannot be easily solved. In this thesis, we mainly focus on network-provider-independent overlays and investigate their performance in providing two different types of service. Specifically, this thesis addresses the following problems:

- **Provider-independent overlay architecture**: A provider-independent overlay framework named Resilient Overlay for Mission-Critical Applications (ROMCA) is proposed. We elaborate its structure including component composition and functions and also provide several operational examples.

- **Overlay topology construction for providing resilience service**: We investigate the topology design problem of provider-independent overlays aiming to provide resilience service. To be more specific, based on the ROMCA framework, we formulate this problem mathematically and prove its NP-hardness. Three heuristics are proposed and extensive simulations are carried out to verify their effectiveness.

- **Application mapping with resilience and QoS guarantees**: Assuming application mapping is the targeted service for ROMCA, we formulate this problem as an Integer Linear Program (ILP). Moreover, a simple but effective heuristic is proposed to address this issue in a time-efficient manner. Simulations with both synthetic and real networks prove the superiority of both solutions over existing ones.

- **Substrate topology information availability and the impact of its accuracy on overlay performance**: Based on our survey that summarizes the methodologies available for inferring the selective substrate topology formed among a group of nodes through active probing, we find that such information is usually inaccurate and additional mechanisms are needed to secure a better inferred topology. Therefore, we examine the impact of inferred substrate topology accuracy on overlay performance given only inferred substrate topology information.

Acknowledgments

My first and most earnest acknowledgement must go to my supervisor Dr. Chris Phillips, who has not only provided academic supervision and support, but also continuous encouragement and experience sharing of both work and life to keep me moving forward.

I would also like to thank Dr. Raul Mondragon, Dr. John Bigham, Dr. Fabrizio Smeraldi and Professor Laurie Cuthbert for their academic discussion and help.

I would like to extend my acknowledgment to my friends in and outside university. Thanks to Dr. Oliver M. Shepherd, Dr. Ling Liu, Ammar Lilamwala, Keith Jones, Iftekharul Mobin, Lexi Xu, Sohaib Qamer, Nadim Mushtaq, Huanlai Xing, Yu Qiao, Xin Chen, Di Bao, Wenxuan Tang, Rehana Kausar, Max Ovidio, Shruti Gopal and many other friends for either their moral support, fruitful discussion and feedback on my work or sharing with me the pressure during my studies. I would also like to thank Professor Mateu Sbert for his academic help as well as stimulating my passion for Spanish.

I want to thank the support staff in our school, especially Kok Ho Huen, Phil Willson, Matt Bernstein, Sue White and Colin Powell, as well as other members of the team for their help with administrative and computer services.

Finally, with my love and gratitude, I would like to thank my parents for tolerating me spending such a long time in school (21 years!) and their continuous encouragement and love, also the parents of my boyfriend for their incessant care and support. Special thanks go to my boyfriend Zhenfei Wang, who is always there to share both my happiness and sorrows during the last three years.

Table of Contents

List of Figures

13

14

16

List of Tables

List of Abbreviations

ALT	Agglomerative Likelihood Tree
AR	Anonymous Router
AS	Autonomous System
BGP	Border Gateway Protocol
Bi-QAP	Bi-Quadratic Assignment Problem
DBT	Deterministic Binary Tree
DDoS	Distributed Denial-of-Service
DDRP	Domain-to-Domain Routing Protocol
DFS	Depth First Search
E2E	End-to-End
EDL-KMST	Enhanced Dual-Layer-aware K Minimum Spanning Tree
FEC	Forward Equivalent Class
FM	Full Mesh
FP	False Positive
GBI	Graph Based Induction
ICMP	Internet Control and Management Protocol
IGMP	Internet Group Management Protocol
ILP	Integer Linear Program
ION	Infrastructure Overlay Network
IP	Initial Pruning
ISP	Internet Service Provider

KMST	K Minimum Spanning Tree
KRC	K Random Connection
LO-MARG	Least Overlap MApping of Regular Graph
LSD	Link State Database
LSP	Label Switching Path
MCMC	Monte Carlo Markov Chain
MLE	Maximum Likelihood Estimation
MLT	Maximum Likelihood Tree
MONET	Multi-homed Overlay NETwork
MPLS	Multi-Protocol Label Switching
MRF	Markov Random Field
MST	Minimum Spanning Tree
NM	Neighbouring Matching
NP-hard	Non-deterministic Polynomial hard
NRA	Non Router Assisted
OB	Overlay Broker
ODON	On-Demand security Overlay Network
ODS	Overlay Directory Service
OG	Overlay Gateways
OLECS	OverLay-based Emergency Communication Services
ON	Overlay Network
OSPF	Open Shortest Path First
OWD	One Way Delay
P2P	Peer-to-Peer
PoP	Point of Presence
QoS	Quality of Service
QSON	QoS-aware Overlay Network
RA	Router Assisted
RC	Random Connection

RNJ	Rooted Neighbour Joining
ROMCA	Resilient Overlay for Mission-Critical Applications
RON	Resilient Overlay Network
RR	Record Route
RTD	Routing Topology Discovery
RTT	Round Trip Time
SA	Simulated Annealing
SON	Service Overlay Network
SP	Simulation Platform
TCP/IP	Transmission Control Protocol/Internet Protocol
TKC	Topology-aware K Connection
TKMST	Topology-aware K-joint Minimum Spanning Tree
TTL	Time To Live
UDP	User Datagram Protocol
VNA	Virtual Network Assignment
VP	Vantage Point
WDM	Wavelength Devision Multiplexing

Chapter 1

Introduction

1.1 Problem Statement and Research Overview

The Internet has now become essential for modern life as a major way to access and exchange information, thanks to the simplicity and flexibility of the TCP/IP protocol suite that it is based upon. At an early stage of the development of the Internet, the best-effort unicast-based service model was sufficient for a variety of applications and services, such as web, file transfer and electronic mail exchange. Its success, together with the advancement of the electronics industry, has in turn stimulated enormous growth [1] and a wider development of network technologies and applications.

With the increasing popularity of the Internet and new applications coming to light, the need for better service other than the best-effort one, such as higher resilience against failure(s), high Quality of Service (QoS), and so forth, is growing. However, it is widely recognized that Internet falls short in meeting additional service requirements [2, 3]. Two major factors limit the efficiency of the Internet in supporting these new require-ments. Firstly, the distributed and autonomous nature of the Internet slows down or even prohibits the process of deploying new features across the Internet to support new requirements. The Internet has now expanded from a small network to a global net-

work consisting of more than 30, 000 Autonomous Systems (AS) [4]. These ASes are administered by different Internet Service Providers (ISP) with their own policies and deployment strategies. Therefore, it is difficult to make fundamental changes to the Internet infrastructure although it would be desirable for supporting new service requirements. Secondly, the *de facto* inter-domain routing protocol deployed across the Internet, i.e. Border Gateway Protocol (BGP), has proved to have many issues that need to be addressed [5]. Although the performance of BGP has been historically acceptable, there are continuing concerns about its ability to meet the demands of the rapidly evolving Internet. For example, the re-convergence time (i.e. the time required for all routers to have a consistent view of the network) can be as slow as several minutes, even tens of minutes [6, 7]. During this period, traffic can be delayed or even get lost. Another example is that the End-to-End (E2E) path selected based on BGP can be sub-optimal [5]. Although there are several proposals to improve the performance of the BGP protocol [7–9], issues such as how to globally deploy an improved version of BGP still needs to be addressed.

In order to meet the QoS and resilience requirements of new services, a couple of approaches have been proposed. The first strategy is to connect to multiple ISPs (i.e. multi-homing), instead of single upstream ISP (i.e. single-homing), to reach the rest of the Internet. There are two models for connecting a multi-homed customer to its providers. The customer network can use one ISP as a primary provider and another as the backup. Thus, the service can still be provided by switching to the backup provider upon detecting loss of path to the primary provider. Alternatively, it can choose to use multiple upstream ISPs and route the traffic to an ISP according to a cost and performance evaluation. Although, multi-homing is a feasible solution, it is still far from being widely deployed because of economical, management and security issues [10]. Moreover, multi-homing cannot remedy the problems facing inter-domain routing.

The second strategy to provide better availability and ensure higher performance is to use an overlay. In overlay networks, traffic can be routed over one or more inter-

mediate overlay nodes according to routing decisions made by periodically monitoring the underlying network. This strategy has several merits. First of all, it does not need any change to the Internet infrastructure, namely, the IP layer only needs to provide basic connectivity between overlay nodes. Thus, it has been considered as one of the promising solutions to facilitate the evolution of the Internet [11]. Secondly, it has the flexibility of path selection according to application requirements and associated network conditions at the time of the decision-making. Usually, end hosts do not have control of the route, thus they will suffer performance degradation in the event of upstream route failure. With the help of overlay networks, end hosts can achieve better performance by using an alternative route provided by an overlay network. As illustrated in Figure 1.1, the source node has multiple routes available to them with the help of the intermediate overlay nodes. Moreover, overlay networks can deal with the slow re-convergence of BGP by reacting much quicker in case of network failures by routing through auxiliary overlay nodes [12]. Thirdly, overlay networks facilitate the deployment and testing of new services, algorithms and technologies. For example, Planetlab [13] is a well-known overlay network with nodes scattered across the world. It has been exploited in many research studies, with regard to multicasting, algorithms for Peer-to-Peer (P2P), overlay
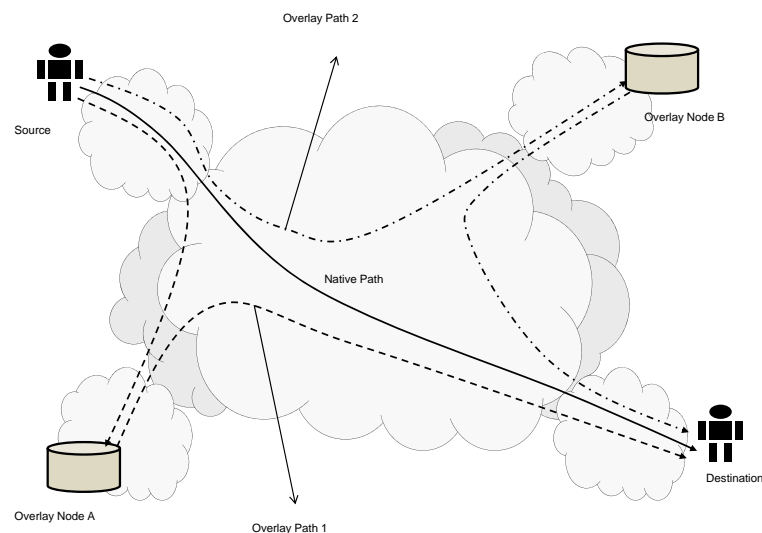
Figure 1.1: Overlay routing architecture

networks and so forth.

As illustrated in Figure 1.2, an overlay involves two layers, i.e. the overlay layer and the substrate layer. In the underlying network, consisting of substrate nodes (including end hosts and routers), packet routing and forwarding are carried out using a traditional routing paradigm. The overlay network sits on top of the substrate network. Namely, several underlying nodes are selected to function as overlay nodes and they are interconnected with each other using overlay virtual links. Each of these overlay links, is actually a path between the two associated substrate nodes in the underlying network. The main advantage of an overlay is the availability of multiple paths between a pair of source and destination nodes. Overlay nodes can route customer traffic using the best path to the destination based on customer requirements bearing in mind the overlay network conditions.

Most of the existing work on overlay networks is aimed at providing better services, such as QoS and resilience, and is focused on network-provider-dependent overlays. Specifically, there is no constraint on where overlay nodes can be placed in the underlying network. Generally, routers possess higher connectivity and thus have greater utility by providing alternative paths with better performance as shown in [14]. However, in multi-domain scenarios, this requires information sharing among different ISPs,
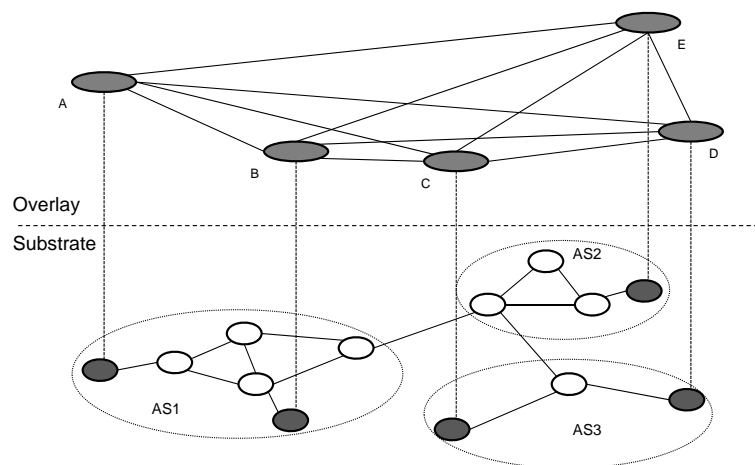
Figure 1.2: Overlay and substrate network layers

who administer their network independently with policies. Therefore, it is difficult, although not impossible, to host overlay nodes across a multi-AS network infrastructure at present. This thesis covers various aspects of overlay network design assuming little support from ISPs, in regard to its ability to provide two different services. The first one is to provide a service with higher resilience so that the customer traffic can still be delivered even though it might not be possible with the traditional Internet alone (It is referred as *"resilience service"* in the rest of the thesis). Facilitated by network virtualization [15, 16], applications with requirements such as resilience and QoS can be fulfilled by strategically choosing a subset of nodes and links for hosting such application requests [16]. The second service we thus focus on is to exploit provider-independent overlays for providing application mapping service with both resilience and QoS guarantees (It is referred as *"application mapping"*[1] in the rest of the thesis.). Moreover, we also investigate the availability and importance of accurate substrate topology information for provider-independent overlays since it is necessary for the satisfactory operation of provider-dependent overlay networks [17] and such information usually needs to be inferred in the case of provider-independent overlays.

1.2 Research Contributions

The contributions made in this study are listed below and the findings of this work can help the design of provider-independent overlay networks offering resilience service as well as application mapping.

1. A network-provider-independent overlay architecture named Resilient Overlay for Mission-Critical Applications (ROMCA) is proposed. The framework exploits a hybrid approach for coping with the network-provider-independence limitation. A centralized component is responsible for service access and overlay topology construction, where appropriate. On the other hand, distributed components are

---

[1]It is also sometimes referred as *"application mapping with enhanced resilience and QoS"*.

cooperatively responsible for service delivery, monitoring and overlay routing functionalities.

2. We formulate the overlay construction with the objective of minimal overlap and prove the NP-hardness of this proposition when ROMCA is used for providing resilience service. Three heuristics are proposed in order to achieve improved resilience in the overlay layer. The effectiveness of the proposed algorithms as compared to existing ones is investigated under various simulation settings.

3. We propose an Integer Linear Program (ILP) model for application mapping with resilience and QoS guarantees. Unlike previous efforts which only try to solve the problem heuristically, the mathematical model can provide the optimal solution, achieving much better QoS and effective resilience.

4. Due to the limitations of the ILP approach for operating with medium/large networks, a novel heuristic is also proposed to find an application mapping solution with enhanced resilience and QoS in a time-efficient manner. Through verification in both synthetic and real-network scenarios, its performance is proved to be better than existing heuristics.

5. A survey of existing methodologies to infer underlying network topologies through active probing leads to a comparison study of the impact of substrate topology information availability and its accuracy on provider-independent overlay performance. Through simulation analysis, we conclude that substrate topology information is important for ensuring good performance with providing application mapping service. Furthermore, appropriate topology inference algorithms are needed to help secure better performance in providing this service.

1.3   Publications

1. Xian Zhang, Chris Phillips,"A survey on selective routing topology inference through active probing", *IEEE Communications Surveys and Tutorials*, **Accepted for publication**, 2011;

2. Xian Zhang, Chris Phillips, "A novel heuristic for overlay mapping with enhanced resilience and QoS", *ICCTA 2011*, **Accepted for publication**;

3. Xian Zhang, Chris Phillips, Xiuzhong Chen, "An overlay mapping model for achieving enhanced QoS and resilience performance", *3rd International Workshop on Reliable Networks Design and Modeling (RNDM'11)*, October 5-7, 2011, Budapest, Hungary;

4. Xian Zhang, Chris Phillips, "Physical-aware topology construction and importance of underlying topological information in provider-independent overlays", *China-Com 2011*, August 17-19, 2011, Harbin, China;

5. Xian Zhang, Chris Phillips, "Construction of provider-independent overlay networks with high resilience", *International Conference on Internet Technology and Applications (iTAP 2010)*, August 21-23, 2010, Wuhan, China;

6. Xian Zhang, Chris Phillips, "On designing the overlay topology of ROMCA (Resilient Overlay for Mission Critical Applications)", *London Communication Symposium 2009 (LCS 2009)*, September, 2009, London, UK;

7. Xian Zhang, Chris Phillips,"Network operator independent resilient overlay for mission critical applications (ROMCA)", *ChinaCom 2009*, August 26-28, 2009, Xi'an, China;

8. Xian Zhang, Chris Phillips, "Network operator independent resilient overlay for mission critical applications", *the annual PostGraduate Network Symposium (PGNet)*, June 22-23, 2009, Liverpool, UK;

1.4   Outline of the Thesis

The rest of the thesis is organized as follows. Chapter 2 provides general background relating to the research elaborated in this thesis and discusses the existing work that is closely related to the material presented in Chapters 3, 4, 5 and 7, respectively. Chapter 3 presents the network-provider-independent overlay framework - Resilient Overlay for Mission-Critical Applications (ROMCA), including its components and corresponding functions and operational examples. Chapter 4 investigates the problem of overlay network construction in the context of the network-provider-independence constraint when resilience is the targeted service for ROMCA. In Chapter 5, an Integer Linear Program model and a novel heuristic are presented to solve the problem of application mapping with both resilience and QoS guarantees. In Chapter 6, a summary of existing methodologies that are able to infer the topology among a group of hosts is provided. Chapter 7 investigates the impact of substrate topology availability and the accuracy of the inferred information on provider-independent overlay performance. We finally summarize the thesis and provide future work in Chapter 8.

Chapter 2

Background and Related Work

2.1 Overview

This chapter first presents general background information relating to overlay networks and their application. Then, we briefly explain the topology characteristics of the Internet which serves as the substrate network for the work presented in this thesis. Thirdly, we provide an overview of the related literature for each theme that is considered. An overview of the main work carried out together with a brief summary of overlay applications is illustrated in Figure 2.1.

2.2 Background

2.2.1 Overlays and their Applications

"A common characteristic of all overlay networks is the existence of two or more layers of networks and the separation of routing and packet forwarding between these two layers" [18]. To be more specific, an overlay consists of a subnet of nodes selected from the underlying network. Overlay nodes are equipped with additional functionalities
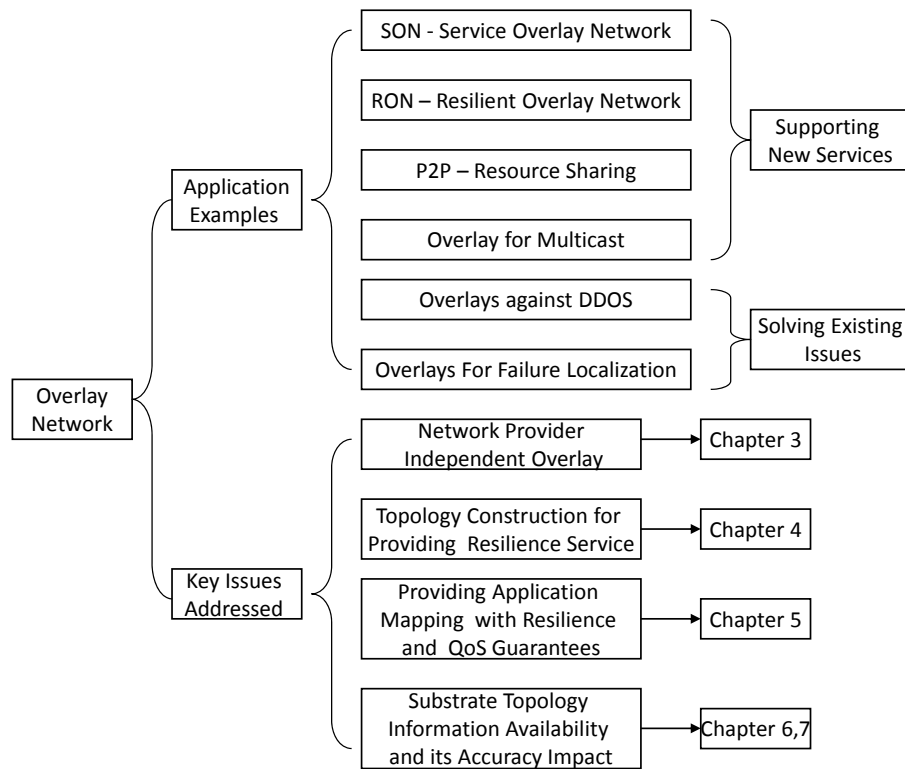
Figure 2.1: A summary of overlay applications and the work presented in this thesis

to support additional feature(s), such as making routing decisions independently from underlying layer(s). In our research, we mainly focus on overlays that are constructed based on the multi-AS Internet, in which the virtual links are composed of IP paths. In this context, we summarize the applications of overlay networks.

Overlay networks are employed either to support new applications that would otherwise be inefficiently provisioned in the current Internet, or to solve existing issues facing the Internet. For either purpose, usually only a subset of the nodes, i.e. end hosts and routers, in the underlying networks is involved.

2.2.1.1 Supporting New Applications

With the rapid development and advances within the telecommunications industry and hardware design technology, the Internet has witnessed widespread development of both

its infrastructure and the services it supports. As the Internet provides "best effort" transmission of data, it exhibits inadequacy in supporting many new emerging applications that demand additional service requirements. Some typical examples of new application requirements, together with a brief discussion of the overlay-based solutions, are given below.

- **QoS-Sensitive Applications**

The goal of QoS measurement is to provide guarantees on the ability of a network to deliver predictable performance. Elements of network performance within the scope of QoS can include latency, error rate and bandwidth. For example, voice streaming and/or video telephony is sensitive to transmission delay. So despite the attraction of the low cost incurred in the Internet, the service quality issue needs to be addressed when moving this kind of service from dedicated networks to an universal best effort infrastructure. Several researchers have exploited overlays on top of the Internet to provide QoS-guaranteed services for this type of application.

J. Shamsi *et al* propose the QoSMap scheme [16, 19] to map an overlay with performance constraints onto the underlying nodes that possess high quality incident links. This is facilitated by network virtualization which make it more flexible in terms of which specific resources to use for a given service request. It assumes the availability of the relevant underlying hop and QoS metrics information.

Similarly, the work of [20] and [21] propose QoS-constraint routing methods in the context of an overlay network. However, these two are based on the assumption that the overlay node locations are pre-determined and have no constraints in terms of where they can be located. The former places emphasis on taking the node computation capability into consideration during the QoS-guaranteed path search while the latter considers the economic cost of the path exploration process. In [22], H. Zhang *et al* discuss and evaluate the performance of an overlay for supporting services with End-to-End (E2E) delay requirements, based on experiments carried out with Planetlab. Their results show

that their overlay has the potential to reduce the E2E delay experienced by end-users by exploiting alternative paths.

- **Fault-Sensitive Applications**

Fault-sensitive applications are referred to those services that cannot tolerate long re-convergence times after one or more failure(s) arise in the network. During the re-convergence process, the affected route may not be available for some time or there may be an attempt to forward the data along another unavailable path. As in the multi-domain Internet situation, different ASes are governed by different ISPs so that the problem cannot be easily solved. Resilience against faults is usually achieved by diverting traffic onto an alternative path in the context of overlays. It can be expressed as a QoS requirement such as the delay variation and/or loss rate experienced by the customers. However, the aim of overlay networks targeting QoS-sensitive and fault-sensitive applications is different. The objective of the former is to find a suitable overlay path within the overlay layer so as to improve the quality of service experienced by end-users, such as minimizing the delay, whilst the latter focuses on the ability of a network to tackle various forms of performance degradation and/or underlying network failures by exploiting possible alternative paths provided via the overlay layer.

Resilient Overlay Network (RON) [23] is the first overlay architecture that is aimed at bypassing the fault using indirect Internet paths through intermediate overlay nodes. It shows that deploying an overlay to tackle the shortcoming of the Internet can provide better resilience against failure(s). A more detailed description of RON is given in Section 2.3.1. The authors in [24] discuss the impact of where the resilience is implemented, i.e. in the overlay or the underlying network, on the network provider revenue and identify the advantages and disadvantages of these two strategies.

The authors in [25] propose OverLay-based Emergency Communication Services (OLECS), which is aimed at providing resilience in the event of AS or multiple inter-domain link failure(s). They deploy overlay nodes at border routers of participating

AS(es) and prove that this can improve the network connectivity and Internet perfor-
mance when exploiting more than three hops in the overlay layer.

- **Multicasting**

Multicasting is an important mechanism for exchanging the information among dis-
tributed, dynamic clusters of heterogeneous nodes for applications such as video confer-
encing, multi-party games. The fact that IP multicasting has experienced difficulties in
global deployment has led to the research of overlay-based multicasting using only end
systems [26, 27]. This is easily deployable. Another advantage is that overlay multicas-
ting can deliver the data among the nodes by deploying a multicast protocol within its
own layer. However, it also has limitations relative to native multicast such as slower
transfer of data.

- **Peer-to-Peer (P2P) Applications**

The overlay-based P2P architecture can provide a good foundation for creating large-
scale data sharing, content distribution and application-layer multicast applications. One
of the main features of P2P networks is that the turn-over of participating nodes is very
high, which means that the node may join and leave the P2P network repeatedly to
download/share the resources with others in the network. A detailed survey is given
in [28].

2.2.1.2   Solving Existing Problems

Besides supporting new applications, the concept of overlays also provides new ways to
tackle existing problems. Fault localization and the detection of Distributed Denial-of-
Service (DDoS) attacks are among some of these applications. In [29], the authors prove
that an overlay-based method can provide a high success rate in detecting failures given
the assumption that there is a reasonable amount of concurrent failures. As for resolving
the DDoS problem, [30] and [31] are two examples that exploit overlays in this research

area. In [30], the authors propose a network-provider-dependent overlay to tackle this issue.

2.2.2 Internet Topology

There is extensive research on how to obtain the topology and associated features of the Internet at various levels [32–34] since it can facilitate better understanding of the evolution of the Internet, improving its performance as well as the services deployed upon it. Although some work shows the power-law distribution of the Internet AS-level graph, there are also some others that contradict this claim [32]. As for the router-level topology, there is no agreed characteristic reported.

In our thesis, we focus on the AS and router-level partial topology formed among a group of nodes scattered across the Internet. Since it is not possible to obtain the details of the true Internet topology, we have mainly employed several typical topologies (either by obtaining the real-network information or using synthetic topologies) that are believed to represent the Internet to some extent to serve as the substrate network. There are many topology generators and real-network datasets available [32] and we only briefly explain the ones that are used later in the thesis:

- **Orbis topology generator**: Orbis [35] exploits real AS or router level topology datasets as input and it is proved to be able to retain the characteristic of the input topology. We mainly use the AS level re-scaled topologies generated by Orbis. However, the publically available AS level datasets are criticized to be incomplete and inaccurate [32]. We thus exploit two different datasets as input, namely the Skitter [36] and the WHOIS [37] AS-level topology datasets.

- **GT-ITM topology generator**: GT-ITM [38] is a popular topology generator and it can generate three types of topology: flat random graphs, N-level and transit-stub graphs. The one usually used by other research related to our work is the transit-stub network because it is believed that this topology can reproduce the

hierarchical structure of the Internet [32]. This type of topology is also adopted in our simulations and verification.

- **Planetlab dataset**: Planetlab [13] is a popular geographically distributed platform for verifying various novel algorithms and techniques involving the Internet as the substrate layer. We have also exploited a dataset provided by iPlane Project [39] with 201 Planetlab nodes actively probing each other.

2.3 Related Work

In this section, we summarize the published research that is closely related to each piece of the work presented in Chapters 3, 4, 5 and 7, respectively. We formally define an overlay, as discussed later in the thesis, as a network with nodes that are either under control of a single administrative entity or cooperate with each other to achieve a common purpose. Usually these nodes are present in the overlay for a comparatively long time. We term this type of overlay as an Infrastructure Overlay Network (ION) so as to distinguish it from a dynamic overlay with nodes only existing in the network for a comparatively short time in order to share resources, such as P2P overlay.

2.3.1 Overlay Architectures

There are several overlay architectures proposed that are closely related to ours in some respects. We introduce them sequentially in order to point out their main benefits as well as weaknesses.

- **Resilient Overlay Network (RON)**

Inspired by the finding of [40] that there are under-exploited redundant paths between pair-wise nodes in the Internet, D. Andersen *et al* [23] propose the overlay network named RON based on an engineering approach. Its main objective is to improve the resilience

and performance of the Internet. Contrary to the high convergence time of BGP, RON can achieve recovery times of the order of tens of seconds through experimental verification. The overlay nodes in RON form a full mesh overlay topology and use both active probing and passive measurement to monitor performance. RON can divert the path to an alternate one if the working path undergoes failure or cannot satisfy the performance requirements. The main interest of their research is to verify the utility of overlays in improving the availability of Internet service by building a test-bed.

In their later work, two other overlay architectures, MONET (Multi-homed Overlay NETworks) and Mayday [2], are proposed. MONET shares the same objective as RON, but it further utilizes explicitly engineered redundant links to address client access link failures. Mayday, on the other hand, is aimed at guarding against denial-of-service attacks by surrounding a vulnerable server with a ring of filtering routers.

- **Planetlab**

Planetlab [13] is a research network built on a global scale that can support the development of new network services including P2P applications, distributed hash tables, network mapping. It is a generalized overlay platform that provides a geographically distributed overlay network capable of verifying new ideas and protocols. The main issues [41] being addressed in the project include: (1) designing the virtual machine running on each participating node; (2) defining and building the management services used to control the test-bed; (3) security and authentication of access to the test-bed services. Till now, it has been used by many researchers to examine the efficiency of their proposed ideas [14, 16, 19, 22, 42–44].

- **Service Overlay Network (SON)**

Z.H. Duan *et al* conceived SON [45], in which each overlay node subscribes a certain amount of bandwidth from its ISP(s). An example of a SON network is QoS-aware Overlay Network (QSON). Z. Li *et al* propose and define the general unified overlay framework composed of overlay nodes termed "Overlay Brokers" (OB) [20]. As described

in their work, there can be one or multiple OBs in a single AS. All the OBs cooperate to perform resource allocation and negotiation, overlay routing and topology discovery. The organization of the OBs is similar to that of the inter-domain routing protocol Domain-to-Domain Routing Protocol (DDRP) [46] extended from Open Shortest Path First (OSPF) [47], namely, it is hierarchically structured. In [20], Z. Li *et al* only focus on QoS routing in the context of the proposed framework. In more recent works of theirs [17, 48, 49], topics like overlay topology construction, overlay failure detection and recovery, and overlay link monitoring technique are discussed respectively.

- **An On-Demand Security Overlay (ODON) for Mission-critical Applications**

The overlay proposed in [50] is targeted at enabling mission-critical communication between personnel at a disaster site and their corresponding agencies that can only be reached via the Internet. The focus is on how they can build an overlay network that can resist failures and external attacks including denial-of-service attacks. Therefore, they focus on the access verification process between the users and the overlay, and the exchange of credentials between the user and the server through overlay transportation. The overlay is built on-demand and one-hop source routing [51] is employed.

In summary, RON has the same purpose as the overlay architecture proposed and described in Chapter 3 in that it tackles the slow re-convergence issue of the multi-area Internet. They focus more on experimental verification of the conjecture that an overlay is helpful in providing better performance relative to the Internet. Conversely, we focus more on issues such as topology construction for providing an efficient resilience service and how to provide an application mapping service with enhanced resilience and QoS. Other related topics such as whether underlying topology information is important and if so, how to ensure the availability and accuracy of underlying topological information is also covered. Planetlab can be used to verify our ideas proposed in the rest of the thesis, as it can provide more practical network scenarios. Unlike the overlay architecture we propose, SON is a typical example of a provider-dependent overlay. Finally, ODON is

listed here for completeness. In terms of overlay application examples, other aspects are covered in Section 2.2.1.

2.3.2   Overlay Topology Construction for Providing Resilience Service

Although the overlays we focus on in this thesis have nodes that persist for a comparatively long time, several examples in the category of P2P overlays for securing more resilient communication are included for completeness.

RON [23] was the first overlay proposed to provide better performance by actively monitoring the network among a group of participating nodes. It boasts a lower reconvergence time of tens of seconds and also the ability to provide an alternative path with better performance as compared to the default Internet path. Nevertheless, the overhead increases of the order of $O(N^2)$, where N is the number of overlay nodes. In [52], the authors discuss the relationship between the effectiveness of an overlay and its overhead consumption. They conclude that a comparable QoS performance to that of full connectivity can be achieved with a lower overlay node degree.

Topology construction in provider-dependent overlays has been researched extensively. For example, in [48], different kinds of overlay topologies are analyzed given overlay node locations. Underlying topology awareness while constructing an overlay topology has been proved to be instrumental in improving the performance of overlay networks. However, customers only use overlay services in the case of direct Internet path failures, which means they should be equipped with the ability to monitor the performance of the original path in a timely manner. Differently, our work focuses on the topology selection issue of network-provider-independent overlays aiming to provide an effective resilience service and we do not expect the customers to have a failure detection ability other than simple interaction with the overlay service provider before using its service described in Chapter 3.

Most existing work on overlays focuses on improving QoS performance using a single

intermediate node for detouring. Moreover, there is no work addressing the resilience of network-provider-independent overlay networks under different failure models. In [53], two availability models for P2P overlays are proposed to find an overlay that can still be fully connected in case of no more than three substrate node failures. They assume that all the substrate nodes are overlay candidates and substrate nodes with a degree lower than three are not considered. Conversely, in our analysis, the overlay node candidates are constrained to be located in ASes with low connectivity as would be expected of smaller tier-3 stub networks. Moreover, their work focuses on proving the NP-completeness of the two models [53] and how to construct an overlay with high availability in a scalable distributed manner [54]. Our work considers finding a good overlay topology solution that is resilient against substrate layer failures/performance degradation. Verification is carried out under different failure models. According to the definition introduced by the authors of [53], overlay nodes can route through all the possible substrate paths between a pair of overlay nodes. However, we assume the paths between the overlay nodes are determined by the substrate layer. It is based on the observation that the nodes in the stub areas generally do not have control over which intermediate nodes they employ for routing purposes unless with the help of other nodes in the overlay layer.

2.3.3 Application Mapping to Achieve Enhanced Resilience and QoS

In various types of multi-layer network such as overlay-over-Internet [16, 43] and IP-over-WDM networks [55–57], how to achieve high resilience has attracted much attention. For instance, Roy *et al* [43] address the issues of how to exploit infrastructure overlay networks to improve the performance of client-server applications. Similar to [16], it is inspired by RON [23]. To be more specific, one-hop intermediate nodes with maximized diversities for all the client-server paths are chosen to increase the resilience of a specific application. In IP-over-WDM optical networks, the resilience issue being researched is termed survivable mapping [55]. In this type of network, the service requests among all

the optical nodes are described by a logical topology. This needs to be mapped onto the substrate layer in such a fashion that a single link failure cannot disrupt the mapped service matrix. Unlike the problem we address later in Chapter 5, only link mapping is considered in these two problems.

There is also much research addressing how to meet the QoS requirements of an overlay request in multi-layer networks. For instance, Service Overlay Network (SON) [45] and Virtual Network Assignment (VNA) both endeavour to provide a QoS-guaranteed service. However, the main research focus in these two cases is how to maximize the revenue and minimize the cost of the network operation. Recently, the work [58] discusses using alternative nodes/links as QoS-compliant backups to reduce the cascading failure effects of single substrate node failures during the overlap mapping process. According to their analysis, shared backup resources can help to improve reliability whilst keeping a lower backup overhead. In terms of resilience, we share the same objective, namely, providing resilience with the least substrate resources possible. However, we also endeavour to attain a QoS-enhanced mapping whilst their objective is to increase revenue.

The most closely related work is the framework QoSMap proposed by Shamsi *et al* [16, 19]. It aims to map a QoS-specified overlay onto the substrate network with the purpose of obtaining guaranteed (and possibly enhanced) QoS performance as well as resilience. Since this problem is NP-hard and is similar to the multi-way separator problem [59], Shamsi *et al* proposed heuristic solutions. In their approaches, QoS performance is enhanced by sequentially selecting nodes with higher quality. Node qualities include the average backup paths a substrate node can accommodate, the QoS quality associated with its links etc. In addition, resilience is provided by specifically allocating backup paths using additional underlying nodes or selected hosting overlay nodes. However, their solutions suffer from two drawbacks. Firstly, they cannot guarantee the best QoS performance since they select potential nodes sequentially and heuristically. The other is that they do not consider the substrate topology when selecting backup paths. Since two seemingly disjoint overlay paths may share common substrate links,

this is insufficient to provide an effective backup. Although sharing the same objectives as QoSMap, we formulate the problem mathematically in order to achieve the optimal solution. Furthermore, we improve the resilience effectiveness by avoiding allocation of backup paths that overlap with their corresponding working paths in the substrate layer for overlay backup use. In order to solve the problem in a time-efficient manner, a new heuristic is also proposed that exploits the substrate topology information and is able to secure effective resilience performance.

2.3.4 Impact of Substrate Topology Information Availability and its Accuracy

Knowing topological information among a group of hosts is desirable or even a necessity for many applications to operate satisfactorily and we summarize the importance of substrate topology information to wider applications/networks. Examples include: (1) network failure diagnosis [60, 61]; (2) network performance monitoring among a group of hosts [62]; (3) applications with a two-layer structure with an upper-layer operating on top of the Internet and the topology information from the lower layer is essential to secure improved performance of the upper layer. Usually, all the participating hosts either operate cooperatively or are assumed to be under control/accessible by a single entity, such as in the work [63, 64]. Since the necessity of topology information is self-evident in network failure diagnosis, we briefly review a number of approaches in the other two categories in this section.

Although such topological information may be accessible in the case of single-domain or private networks [65], it is typically difficult to obtain for inter-provider networks. The reason lies in the fact that ISPs, which administer their own networks independently, seldom make the intra-network information publicly available for security and business management reasons [32]. Moreover, although it can be argued that the topology among a group of hosts can be gathered by extracting relevant topological details from the publicly available Internet topology information [66], this is not desirable for several reasons.

Firstly, the publically available Internet topology information is usually anonymized for security reasons. Hence, it is difficult to identify the networks that are of interest. Secondly, it suffers from the sampling bias problem[1] and may not accurately represent the characteristics of the part of the Internet that is under consideration. Finally, the Internet is highly dynamic [32] so the published topology information can quickly become inaccurate or obsolete. Hence, for those activities spanning multi-domain networks, inference of the routing topology among the group of hosts involved in the activities mentioned in this section is required.

A typical example that exploits the network topology is the inference of link-level metrics from path-level metrics obtained through E2E probes. It is of considerable interest for both ISPs as well as Internet users to monitor their network performance for network engineering purposes so they can better control the network or adjust their behaviour accordingly. Ideally, the relationship between these two metrics is expressed using the tomography equation as shown below [67]:

$$Y_{m,1} = G_{m,n} \times X_{n,1} \tag{2.1}$$

where $Y_{m,1}$, $G_{m,n}$ and $X_{n,1}$ represent the E2E path-level metrics matrix, the routing matrix and the link-level metrics matrix, respectively. Furthermore, the dimension of the matrix is notated using m and n, which denotes the number of end-to-end paths concerned and the number of the links in the network respectively. As shown in Equation (2.1), in order to obtain the link-level metrics, topology together with the routing information is necessary. It is easy to infer from (2.1) that inaccurate topological/routing information will result in an inaccurate estimation of link-level metrics.

A second example taking advantage of routing topology among a group of hosts is infrastructure overlay topology construction, where all the nodes are dedicated entities to provide better resilience/quality-of-service by exploiting application-layer routing. Sev-

---

[1]This problem arises because topology inference methods usually obtain the traceroute information from a subset of nodes in the Internet, namely by sampling information from the Internet. This is further explained in Chapter 6.

eral researchers [43, 48, 68] have proved that an overlay topology that considers the characteristics of underlying topology can perform better than those that simply ignore the underlying topology information. For example, Zhi Li *et al* [48] propose two heuristic substrate-aware overlay topology construction algorithms and verify their superiority over other methods under with network scenarios. This is achieved by avoiding selecting paths that overlap to a greater degree with each other in the substrate network. However, no research to-date has provided a discussion on the impact of underlying topology availability and its accuracy on overlay topology construction if such topological information is important but can only be inferred.

Another example concerns algorithms that attempt to obtain E2E network performance data with reduced probe traffic being injected into the network. For instance, if end-to-end delay information along the paths between host pairs is required then pairwise probing is usually needed to achieve this. However, since E2E performance needs to be monitored regularly, a large volume of probe traffic is incurred if pairwise probing is adopted. If the delay along a path between two hosts can be obtained by indirectly exploiting the delay information of other paths between several host pairs, it may be unnecessary to inject probe messages for this host pair. In brief, probe traffic reduction can be achieved by taking advantage of the correlation of the end-to-end paths formed among a group of nodes. A couple of works have discussed this given the assumption that the underlying substrate topology is known. For example, Yan *et al* [64] propose reducing the extent of virtual path monitoring by exploiting tomography principles. However, inaccurate information concerning the underlying topology deteriorates the performance of the proposed method as indicated in their discussion. Moreover, the accuracy of the topology and corresponding routing matrix, as analyzed in [69], does have an impact on the accuracy of overlay path metric prediction. Chiping *et al* [63] propose an orthogonal method by trading estimation accuracy with a lower probing overhead. They bound the additive/bottleneck metrics of virtual links by exploiting a limited number of overlay probes instead of aiming to obtain the most accurate information in order to reduce the

overlay probe cost. As with Yan's work, their method is based on the hypothesis that the underlying topology is known to the overlay. If only partial topological information can be obtained, the estimation accuracy will be degraded, accordingly.

As discussed in this section, substrate topology information is important for many applications including overlay networks. Therefore, we provide an overview of existing methodologies for inferring the selective topology among a group of hosts in Chapter 6. Based on this, we discuss the impact of topology information availability and its accuracy on provider-independent overlay performance in Chapter 7.

2.4   Summary

In this chapter, we have provided background information and a review of the closely related work to each element of our research. Each aspect is summarized and discussed to differentiate our work from the existing state of the art.

Chapter 3

A Provider-independent Overlay Architecture - ROMCA

3.1 Overview

This chapter first provides the motivation for proposing a network-provider-independent overlay named Resilient Overlay for Mission-Critical Applications (ROMCA). Then, we specify the two applications we focus on that make use of the ROMCA framework and describe the proposed overlay architecture in depth. Finally, a couple of operational examples are presented for each application.

3.2 Motivation

As summarized in Chapter 2, overlay networks can improve the performance of the Internet such as providing higher availability as well as facilitating new service deployment. However, the inter-provider trust issue is one of the main problems that face most of the existing overlay architectures that need the cooperation of multiple providers and cannot be solved in a short term. Therefore, we propose a provider-independent over-

lay architecture named Resilient Overlay for Mission-Critical Applications (ROMCA), assuming little support from network providers.

The performance of the ROMCA overlay will typically be inferior to that of network-provider dependent overlays. For example, it cannot recover from access link failures for a given overlay node, i.e. the first substrate link the overlay node employs to reach the rest of the network. This is because this overlay node will lose all its connections to the rest of the network if anything happens to its solitary access link. A feasible solution is to use multi-homing to avoid single-points-of-failure for overlay nodes. Another limitation relates to the QoS performance. For example, the delay of an overlay path is the sum of the delay of all the substrate links it comprises and the bandwidth of an overlay path is that of the bottleneck link in the set of substrate links covered by this overlay path. As there are no overlay nodes with rich connectivity present in ROMCA, the path performance that can be attained will be inferior to that of provider-dependent overlays. However, the ROMCA architecture avoids the inter-trust problem and it can still provide better resilience as compared to that of the Internet. Moreover, it offers a transitional solution to applications that operate between multiple parties, which require better resilience than that of the native multi-domain Internet. Moreover, the ROMCA framework is also instrumental in setting up a private overlay network under a single administrator for resilient communication across multiple geographical locations or applications that would be otherwise inefficiently accommodated by the traditional paradigm of the Internet.

3.3 ROMCA Framework

3.3.1 Target Applications

There are many applications that a provider-independent overlay can support as discussed in Chapter 2. As indicated by the name, ROMCA focuses on providing higher

resilience. In the ROMCA framework, we focus on provide better resilience for two different types of applications. The first one is providing resilience service for applications that require better resilience from the service-providing network than that can be provided by the Internet. The second one is application mapping for achieving enhanced resilience and QoS. We refer the two applications as *resilience service* and *application mapping* in the rest of the thesis. Since ROMCA has the flexibility of choosing the paths within its own layer, it can provide better resilience by exploiting the alternative paths via intermediate overlay nodes. Therefore, it can fulfill the requirement of both applications, which would not be sufficiently supported with the traditional service paradigm of the Internet.

3.3.2 ROMCA Architecture

In this section, we explain the ROMCA architecture from both component and functional perspectives. Moreover, we provide a brief summary of the key research points within this proposed framework.

3.3.2.1 Component Composition

As shown in Figure 3.1, ROMCA consists of Overlay Gateways (OGs) and an Overlay Directory Service (ODS). The ODS is a centralized functional module that is used to manage the resources in the overlay and provide service access. One example of the first function is that it gathers partial substrate topology information obtained by all the overlay nodes through active probing and constructs the substrate network topology for overlay use. For example, in order to provide resilience service with reduced overhead, the ODS can form a partial mesh overlay topology exploiting the acquired substrate network topology as depicted in Figure 3.1. Another example is to exploit the obtained topology as well as QoS performance of the substrate network to facilitate selection of a suitable subset of overlay nodes and links to undertake the application mapping service.
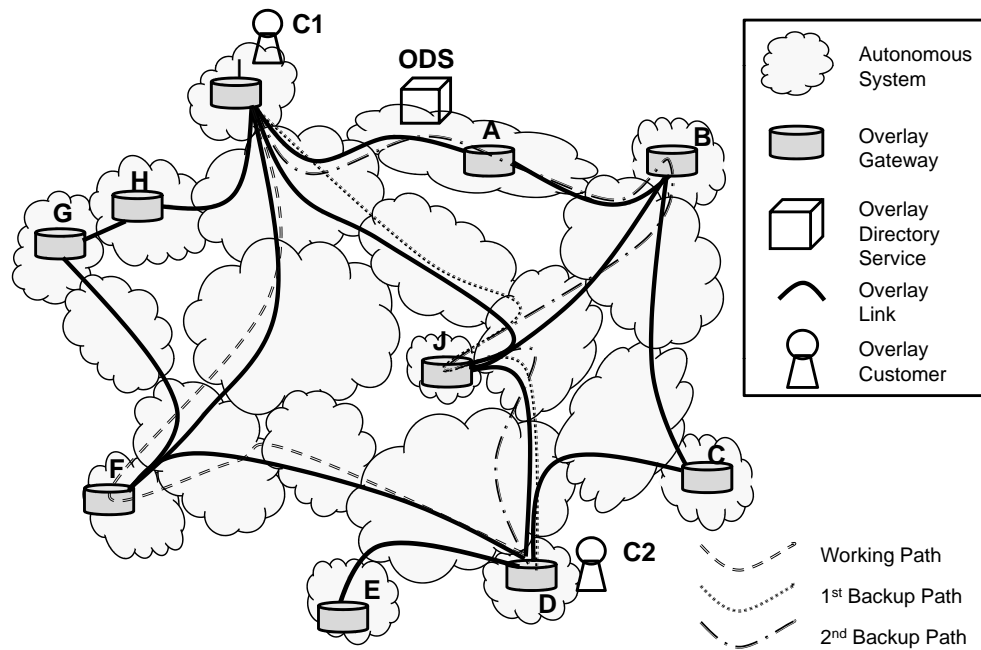
Figure 3.1: The ROMCA architecture

However, it plays no part in the actual forwarding of traffic across the overlay. On the other hand, OGs are the entities that deliver the traffic and are responsible for the following functions:

- Neighbour connectivity probing and monitoring, performance information exchange between adjacent OG(s);

- Routing and performance information collection and dissemination in the overlay layer;

- Service provisioning, including establishing, maintaining and removing working and backup paths for customer traffic;

- Resilience-related functions, such as failure notifications to other OG nodes and the ODS.

Together the single ODS and multiple OG entities form the ROMCA architecture and are the means by which ROMCA provides resilience service as well as fulfilling

application mapping service. The architecture itself can be effectively hidden from the end-users, which simply know the public address of the ODS from which the nearest OG point-of-presence is obtained.

3.3.2.2 Functional Composition

As depicted in Figure 3.2, there are five basic functions in the ROMCA framework and one service-specific function block for each of the applications we are focusing on.

- **Monitoring/probing**: This function is a basic function of the ROMCA overlay and it is used to monitor the performance as well as detect the failures in the substrate network the overlay sits upon. This function is undertaken by all the OG nodes in ROMCA.
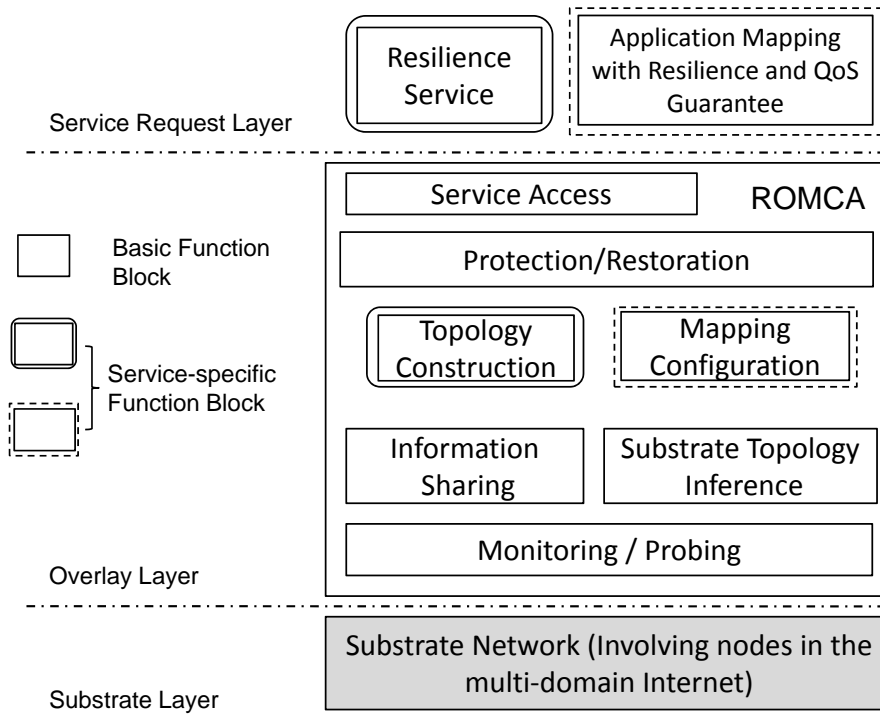
Figure 3.2: The functional composition of the ROMCA overlay

- **Information sharing**: This function is to share information across the overlay layer. A distributed mechanism for topology updating and network performance information flooding is needed so that OGs and the ODS can maintain up-to-date performance information to enable them to efficiently establish working and backup paths for customer traffic. In our architecture, a flooding mechanism similar to that used in OSPF [47] is deployed among the OGs. Therefore, update packets will be flooded to all OGs and the ODS when the performance of a virtual link changes across a threshold and they can store the updated information in their Link-State Database (LSD).

- **Substrate topology inference**: This function is fulfilled cooperatively by the OGs and the ODS. All the OG nodes report their partial topology information obtained through active probing such as tracerouting entries to the ODS and the ODS is responsible for constructing a complete view of the part of the substrate network that is of concern to the overlay.

- **Protection/restoration**: Protection/restoration techniques can help ROMCA switch the traffic to an alternative path within its own layer and thus to maintain service provisioning capability. In ROMCA, a mechanism similar to Multi-Protocol Label Switching (MPLS) [70] is used to define the working and protecting Label Switched Paths (LSPs). There are many well-established protection and restoration strategies [71] proposed for MPLS-based networks that can be exploited in ROMCA. A concrete example is provided later in this chapter to explain this in further detail.

- **Service access**: This is the interface ROMCA provides to negotiate with customers and accept customer requests.

- **Topology construction**: This function is required to construct an efficient overlay topology so as to provide a resilience service with reduced overhead. A simple example would be to construct a full mesh overlay topology as in [23].

- **Mapping configuration**: This function is needed for mapping an application request with specific resilience and QoS requirements. It includes mapping the virtual nodes and links of the application request as well as provisioning a backup path for each application link.

3.3.2.3 Key Research Points in the ROMCA Framework

In the ROMCA framework, we focus on the following key aspects:

- **Overlay topology construction for providing resilience service**: Although, ROMCA shares the same objective of promoting resilience across the Internet using alternative node(s) such as RON [23], we aim to achieve this by employing a partial-mesh overlay topology that can guarantee the resilience of working and backup paths to some extent. This can reduce the overhead incurred in the overlay and thus ROMCA is more scalable. This is addressed in Chapter 4.

- **Application mapping with resilience and QoS guarantees**: Previous work on this problem does not consider the impact of the substrate topology in achieving resilience. We take the initiative of incorporating this factor into application mapping facilitated by the substrate topology inference function built in ROMCA. This is addressed in Chapter 5.

- **Substrate topology inference**: Although there are works addressing how to obtain the whole Internet topology [32–34], there is no summary of how to infer the topology formed among a specific group of nodes. Thus, we summarize all the techniques that can be exploited for fulfilling this function. Since ROMCA is provider independent, we mainly focus on the active-probing-based methods as summarized in Chapter 6.

- **The impact of substrate topology availability and its accuracy**: Resilience provided in the overlay layer is dependent on the topology features of the substrate

layer the overlay is deployed upon. Although we assume such information is given in Chapter 4 and 5, it is not usually available for overlay exploitation based on the summary in Chapter 6. Therefore, we extend our analysis of the provider-independent overlay performance to scenarios where the substrate topology can only be inferred. This analysis is described in detail in Chapter 7.

3.4 Service Provisioning and Operational Examples

In this section, we first provide a simple service provisioning example for each application. Then, focusing on providing resilience service, we explain operational examples and describe the overlay node setup process.

3.4.1 Service Provisioning Examples

Figure 3.3 and Figure 3.4 depict how ROMCA provisions a resilience service request and an application mapping request, respectively. As shown in both examples, the resilience is provided by setting up a working/backup path pair so that the service can be quickly recovered by the backup paths if there is any problem with the assigned working paths.

In the next subsection, we focus on elaborating the operation procedure for providing the resilience service. As for the deployment of the application mapping with resilience and QoS requirements, the ODS determines the mapping of both nodes and links for an application request according to the requirements of the application and the condition of the overlay network. Since the essence of providing resilience for application mapping is similar to that of the resilience service, it is not explained further.
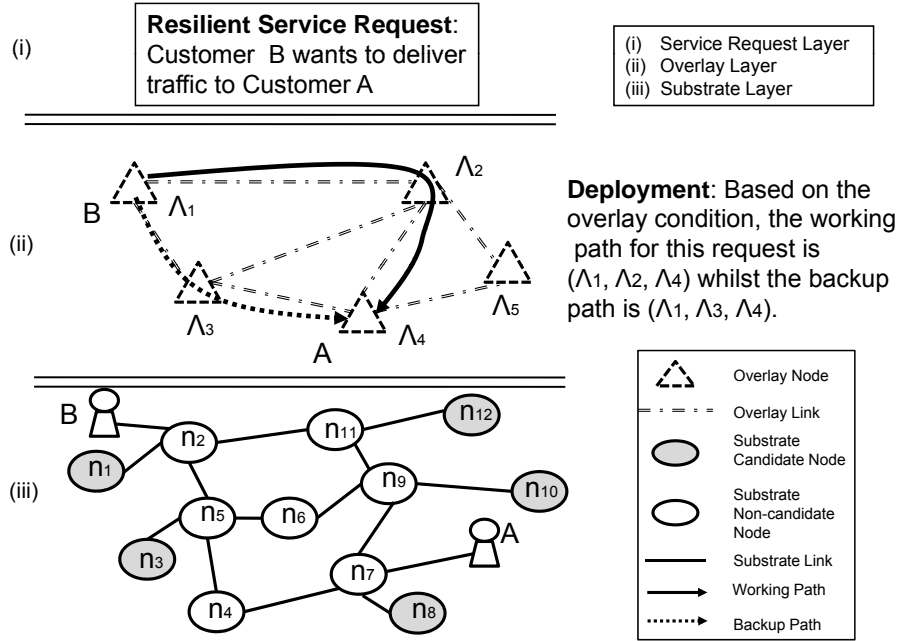
Figure 3.3: An example of deploying the resilience service
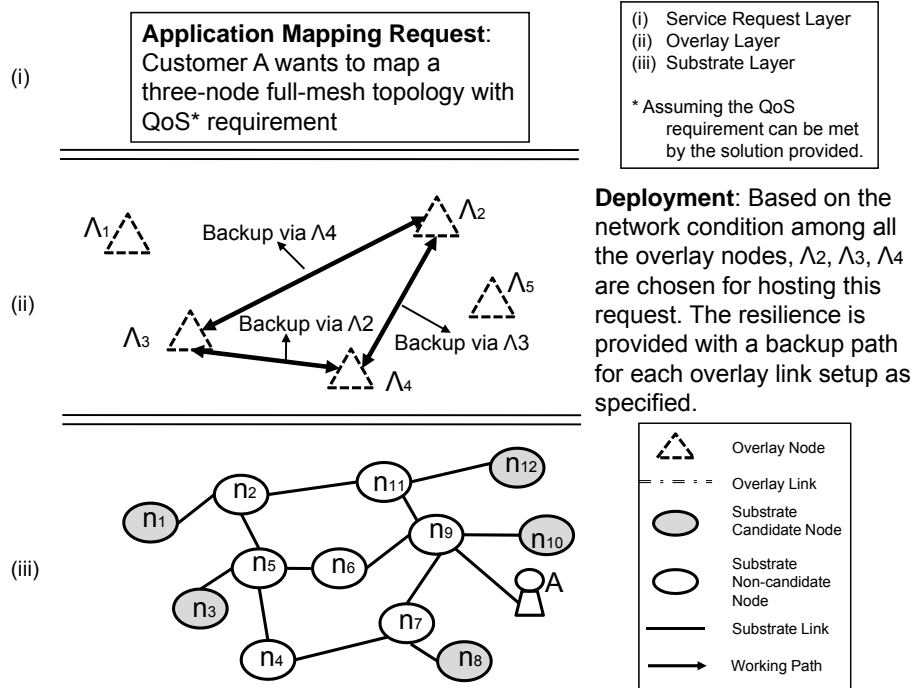
Figure 3.4: An example of deploying application mapping with resilience and QoS guarantees

3.4.2   Operational Examples for Providing Resilience Service

To explain the service provisioning process for providing the resilience service, consider the topology depicted earlier in Figure 3.1, where a mechanism similar to MPLS [70] is used to define the working and protecting LSPs and is illustrated in depth. Most of the existing proposed overlays, such as the work of Zhi Li *et al* [48] and S. Roy *et al* [43], assume customers have their own way of detecting anomalies on their direct routes to the destinations and use the overlay service as a backup if anything happens on their direct routes. However, additional functionalities are needed on the customers' side. Differently, in ROMCA, we assume that customers send their service requirements directly to the access point of our overlay network and obtain the service they want according to the agreement made with the ROMCA. ROMCA can provide both working and backup paths for its customers.

Consider a ROMCA customer, i.e. customer 1, located in the same Domain as OG I. The customer approaches the ODS providing the IP address of itself and the IP address of the destination customer (i.e. customer 2) from which the ODS can infer their proximity to various OG nodes that are operational. The whole procedure is illustrated in Figure 3.5 with a step-wise explanation in Table 3.1.

In this example, the ODS knows the customer has no desire to become an OG; it simply wishes to exploit the overlay mesh to provide resilient pathways to a fellow customer in another AS. The ODS provides it with the nearest point-of-presence, i.e. the address of OG I, and a service "ticket" that it needs for using the service. The ODS also informs the local OG, i.e. I, that it can expect an approach from customer A and its requirements as well. The "ticket" is a unique identifier for a specific service request for a customer and it also serves as a means for the ingress OG to verify the approaching customers as well as setting up corresponding working paths using MPLS and possibly backup paths according to the service requirements.

When customer 1 contacts OG I with the information provided by ODS, I checks the
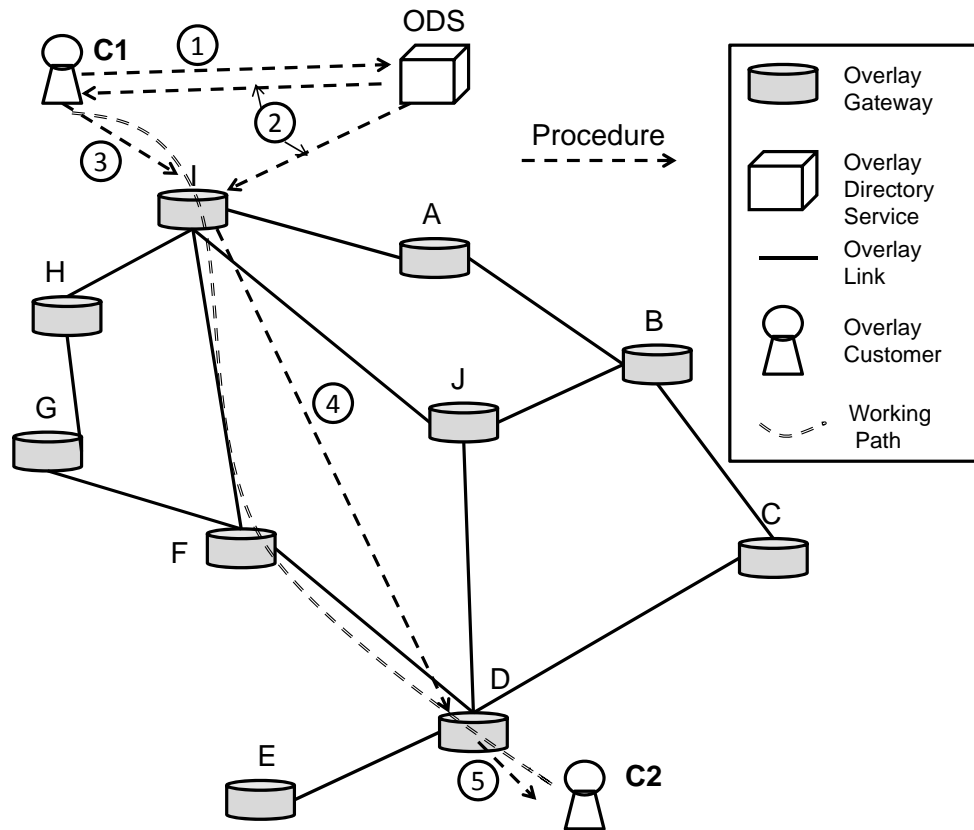
Figure 3.5: An operational example for providing resilience service

ticket details. In this case, the customer wishes to establish disjoint resilient paths to fellow customer 2. Using its LSD, OG I sends RSVP path-establishing messages to OG D using diverse links and nodes where possible. For example, the working path may be via OG I, F and D.

The corresponding protection path could be: OG I, J and D. Once established, a FEC[1]-to-Label binding entry is created at OG I, and customer 1 is informed that the service is ready. Traffic from customer 1 to 2 now uses IP to reach OG I (using IP tunnelling). Accordingly, the packet has an MPLS label pushed onto it and this is encapsulated in a datagram for OG F. At intermediate OGs, the MPLS shim layer is examined and the label is swapped and the traffic re-encapsulated and sent to the next-hop OG, and so on. This flow of label switching is depicted in Figure 3.6. At node D,

---

[1]A FEC is used in MPLS to identify the traffic with similar characteristic and here it is used to denote traffic request from one customer associated with a specific communication session.

Table 3.1: Explanation of the operational example for providing resilience service

Step No.	Procedure Name	Explanation
1	Service Requesting	Customers send service requests with specifications to the ODS using its publically known IP address.
2	Service Preparation	(1)Customers obtain the unique identifiers as well as the IP address of the ingress OG node; (2) The chosen ingress node will get the customer verification information as well as customer request information from the ODS. Then, it sets up the overlay path ready for traffic forwarding using signalling protocols such as Resource Reservation Protocol (RSVP) [72].
3	Ingress Initiation	The customers initiate the traffic delivery process by sending packets to the ingress node with their own identifiers. The ingress node will verify the approaching customers by comparing the identification number provided by the customers and the ODS. If matching, then it will deliver the traffic sent over from the customers; otherwise, it will reject the request.
4	Traffic Forwarding	The packet will be delivered over the established label switched overlay path. When a node receives a packet, it will inspect the label to determine the next-step overlay node and re-encapsulate the packet using next-hop IP address and label number.
5	Traffic Delivery	The egress OG node strips the additional information used by the overlay and delivers the traffic to the destination using normal IP packet formatting.

the label is popped and the datagram delivered to customer 2 as per standard IP.

If a failure or mis-configuration happens in a transit AS, it may take the ASes several minutes to re-converge and thereafter find the proper route to divert the traffic accordingly. But in ROMCA, as the neighbouring OGs exchange "hello" messages periodically (e.g. tens of seconds), the failure will result in loss of these heartbeat messages. Once the time threshold for neighbouring connectivity loss is reached, the OG(s) adjacent to the point(s) of failure will propagate the information over the virtual links to the ingress point(s), which can immediately update the FEC-to-Label binding so that the traffic is
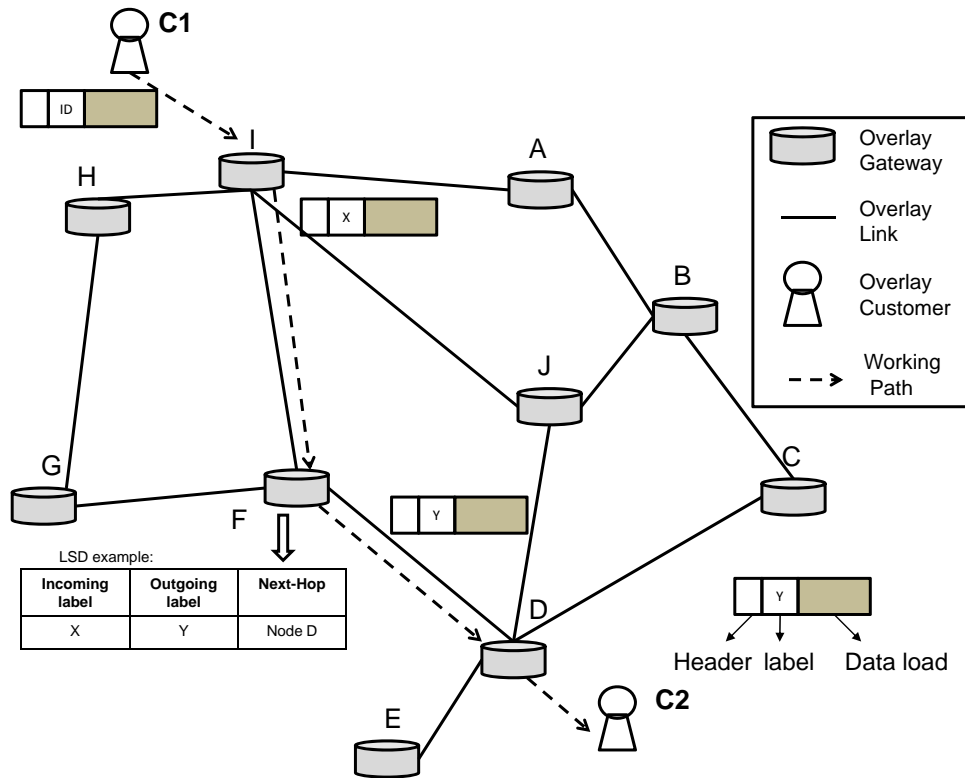
Figure 3.6: Label switching process during packet delivery

mapped onto the pre-configured diversified protection path(s). These paths avoid the "failed" AS and so ensure that service delivery is quickly re-established.

As the service provisioning example shows, when ROMCA is used, traffic from customer 1 to 2 goes via the ingress OG I, and from there, follows a working path dictated by the LSP. Moreover, a protection path is also set up for resilience purposes. In the event of a failure in an AS lying between OG I and F, the ingress OG will switch the traffic onto the protection LSP, i.e. to the path going from OG I, J to D, thus re-establishing customer data packet delivery typically within tens of seconds.

Another example is the dynamic mapping of LSPs according to the updated monitoring results. If the virtual link between I and J results in a longer delay than that of the path from I, via A, B to J, the backup LSP can be dynamically changed to the alternative backup LSP as depicted in Figure 3.1, while the working LSP remains unchanged.

3.4.3 Overlay Setup Process for Providing Resilience Service

Overlay topology construction is required for providing the resilience service whilst the overlay topology is determined by the requests in application mapping service. In this section, we focus on explaining the overlay setup process assuming a substrate node applies to join ROMCA.

It can be seen from Figure 3.1 that the overlay topology is partially meshed and generally organized into inter-connected cycles, though stub connections are permitted. There are various ways that ROMCA can be constructed: (1) a fixed group of OG nodes dedicated to a common objective of providing resilience services among themselves or as a type of value-added service to wider customers; (2) all of the OG nodes as volunteered nodes where they can obtain benefit from joining the ROMCA network, such as free resilient services; (3) A combination of (1) and (2), where both types of nodes co-exist. The main issue discussed in Chapter 4 is how to build a resilient overlay topology given a certain number of dedicated nodes. In the rest of this section, we mainly illustrate the node joining mechanism for the last two modes where there are volunteering nodes wishing to become OG nodes.

An example of the basic node joining process for new nodes is depicted in Figure 3.7. As explained previously, a distributed mechanism similar to OSPF is deployed for topology updating and network performance information flooding so that OGs can maintain up-to-date performance information to enable them to efficiently establish working and backup paths for customer traffic. In our architecture, update packets will be flooded to all OGs when the performance of a virtual link changes across a threshold and they can store the updated information in their Link-State Database (LSD). This in turn will influence the routes chosen by the ingress OGs for subsequent working and backup paths.

The virtual links between adjacent OGs nodes are chosen according to probing results and network performance measurements. For instance, assume node G applies to the ODS to join the overlay with the topology depicted in Figure 3.1. After retrieving the
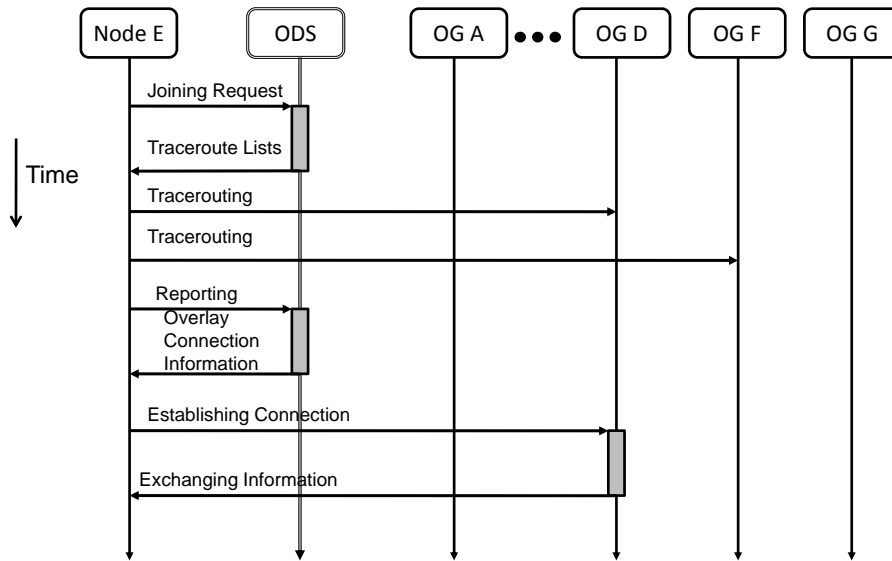
Figure 3.7: The node joining process (using node E as an example)

potential neighbouring information from the ODS and trace-routing to these corresponding nodes, G reports its findings to the ODS. It is accepted into the ROMCA topology as it is a "valuable" transit node having Layer-3 diversified paths to H and F, i.e. from it multiple exit points can be reached from its local AS .

There are also situations where stub nodes may be accepted according to their resilience and service access utility. For example, node C is viewed as a potential alternative path for B and D, so there are two virtual links connected from C. Whereas, the stub node E is only accepted into the overlay by establishing one virtual link connected to D. Node E is simply used as a service access point for the customers residing in its own AS.

3.5 Summary

In this chapter, a network-provider-independent overlay architecture called ROMCA is proposed in order to provide better service across a multi-domain environment for mission-critical applications including resilience service and application mapping. First,

its composition and functionalities are explained. Then, details of the service provisioning for both applications and examples of the operational procedures and overlay setup process for providing resilience service are provided to complete the explanation of the proposed overlay architecture.

Chapter 4

Topology Construction for Providing Resilience Service

4.1 Overview

This chapter focuses on the topology construction of the ROMCA overlay with the objective of providing resilience service. In order to achieve high resilience, we formulate the topology construction task as an optimization problem. Since it is analogous to the Bi-Quadratic Assignment Problem [73] which is NP-hard, three heuristics are proposed and evaluated.

4.2 Problem Description

In our discussion and analysis in this chapter, there are two network layers involved, one running on top of the other. An example of such a two-layer network is illustrated in Figure 4.1. Moreover, Table 4.1 summarizes the notations used throughout this chapter.
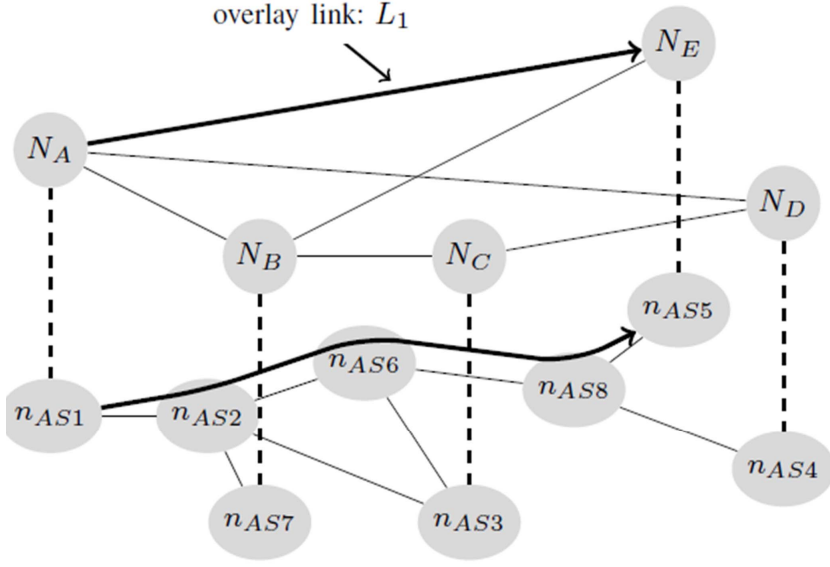
Figure 4.1: A two-layer network example

As depicted in Figure 4.1, the overlay network is placed upon the underlying network. Therefore, an overlay link between node i and node j can be expressed as $L_P = p_{i,j} = \{n_i, \ldots, n_j\}$. For example, in Figure 4.1, $L_1 = p_{n_{AS1}, n_{AS5}} = \{n_{AS1}, n_{AS2}, n_{AS6}, n_{AS8}, n_{AS5}\}$. Depending on the type of underlying networks that are involved, i.e. AS-level or router-level, an overlay link consists of either an ordered AS-node [1] or router-node list. We use both AS-level and router-level topologies as the substrate network. Since the problem formulation is independent of the substrate network type, we focus on using the AS-level underlying topology for formulating the problem.

Two overlay links can share more than one AS and the overlap is defined as the number of overlapped ASes and can be expressed as:

$$OL(L_I, L_J) = OL(p_{a,b}, p_{p,q}) =\mid p_{a,b} \cap p_{p,q} \mid \qquad (4.1)$$

where $L_I = p_{a,b} = \{n_{AS_a}, \ldots, n_{AS_b}\}$ and $L_J = p_{p,q} = \{n_{AS_p}, \ldots, n_{AS_q}\}$ respectively.

Since a failure in the underlying layer can trigger one or multiple failures in the overlay, it is of crucial importance for an overlay to retain as many connections as

---

[1]Here, the node is an abstract representation of an AS.

Table 4.1: Notations and definitions

Notation	Definition
The overlay network	
$G_O(V_O, E_O)$	This describes the overlay topology.
N_I, L_J	Overlay nodes and links
$|N_I|$	Overlay node number
C_o	Overhead incurred in the overlay layer
$P_{I,J}$	A path in the overlay, $P_{I,J} = \{N_I, \ldots, N_J\}$
$OL(L_I, L_J)$	The overlap between two overlay links and it can be measured in link/nodes they share in the corresponding paths in the substrate layer.
K_I, K_{AVG}	The node degree of an overlay node N_I and average node degree in the overlay
K_{MAX}	The maximum node degree constraint in the overlay
$\Theta(I, J)$	(Binary)=1 if there is a route between N_I and N_J; otherwise 0.
W_{L_I}	The weight of a link in the overlay
ρ	Overlay network resilience
ρ^*	(Benchmark) The optimal overlay network resilience achieved by the Full Mesh overlay construction algorithm
The substrate network	
$G_s(V_s, E_s)$	This describes the substrate topology.
n_i, l_j	Substrate nodes and links
k_i	The node degree in the substrate layer
$p_{i,j}$	A path in the substrate layer, $p_{i,j} = \{n_i, \ldots, n_j\}$ and $|p_{i,j}|$ denotes the number of nodes included in this path.
k_{max}	The maximum node degree constraint in the substrate layer
For simulated annealing	
$G_{reg}^{K_{MAX}}$	A regular graph where all nodes have degree equal to K_{MAX}
$T_{initial}$	Initial temperature
M_j^I	(Binary)=1 if the overlay node N_I is mapped to the substrate node n_j; otherwise 0.
$f_{Ij,Km}()$	Swapping function, in which the mapping relationship of two overlay nodes is exchanged. Namely, N_I and N_K will be mapped to n_j and n_m respectively if N_I is originally mapped to n_m and N_K to n_j .
$V(t)$	An overlay construction solution at time t.
$V_{temp}(t)$	A temporary new overlay construction solution at time t
N_{try}	(Constant) iteration number for certain temperature
N_{max}	(Constant) number of rounds for the cooling process
μ	The cooling rate controlling the cooling speed

possible in the presence of failures in the substrate layer. Therefore, we use the resilience of an overlay network as the evaluation metric and it is defined as:

$$\rho = \frac{\sum_S \sum_{D,D \neq S} \Theta_f(N_S, N_D)}{\sum_S \sum_{D,D \neq S} \Theta(N_S, N_D)} \tag{4.2}$$

where the sum is for all source-destination pairs and $\Theta_f()$ denotes if a route exists after the failure of one or more ASes in the underlying layer.

The objective of overlay topology construction for providing resilience service is to find an overlay topology $G_O(V_O, E_O)$ that can have maximum resilience under various underlying network failure(s) scenarios. However, it is difficult to predict the failure(s) that occur in the Internet. Moreover, there are no known accurate failure models that can adequately describe failure(s) that take place in the Internet. Suggested by the work done by Zhi Li *et al* [48], to construct an overlay network with maximum resilience can be stated as being equivalent to designing an overlay network with a higher number of substrate node/link-disjoint virtual links. They proposed a heuristic by constructing minimum spanning trees sequentially complying with the overlay degree constraint. Instead, we formulate it mathematically so as to minimize the overlapping among all the virtual links, which can be formally expressed as to minimize:

$$\sum_I \sum_{J,J \neq I} OL(L_I, L_J) = \sum_{s,d} \sum_{s',d'} |p_{s,d} \cap p_{s',d'}| \tag{4.3}$$

for all overlay node pairs. Given two overlay nodes N_I and N_J, we need to decide whether to assign an overlay link L_M to the path connecting these two nodes or not.

If $\theta(I, J)$ is binary and denotes whether there is a link established between N_I and N_J or not, then $\theta(I, J)M_s^I M_d^J = 1$ denotes that the N_I is mapped to n_s and N_J to n_d and the overlay link between them is setup (i.e. it is mapped to $p_{s,d}$). Thus, the

objective stated in (4.3) can be rewritten as:

$$\sum_I \sum_{J,J \neq I} OL(L_I, L_J) = \sum_{I,J,K,L} \sum_{m,p,s,t} \theta(I,J)\theta(K,L)OL(p_{m,p}, p_{s,t})M_m^I M_p^J M_s^K M_t^L \quad (4.4)$$

which is a Bi-Quadratic Assignment Problem (Bi-QAP) [73].

4.3 Proposed Substrate-aware Algorithms

It is known that the Bi-QAP problem is NP-hard [73] and there is no known way to find the optimal solution to this problem in polynomial time even for a network of moderate size. Since the problem is intrinsically difficult, two substrate-aware heuristics are proposed and described in this section [2]. Before presenting these two heuristics, the constraints associated with network-independent overlays are explained.

4.3.1 Constraints

In order to place the proposed model in context of our ROMCA overlay architecture, there are a couple of constraints that we need to consider.

- **Restriction I:** is on the degree of the ASes that overlay nodes can be placed in, i.e. $k \leq k_{max}$. This constraint means the nodes that will apply to become an overlay node will only be those nodes in the ASes with lower connectivity (i.e. typical of tier-3 networks and lower). This constraint can represent the network-provider-independence feature of the proposed overlay.

- **Restriction II:** as analyzed by Zhi Li [48] for overlay topology construction in provider-dependent and single-domain cases, is the overhead C_o introduced by overlay monitoring and probing. Given a fixed number of overlay nodes, prob-

---

[2]The third heuristic that does not need substrate topology information is present in Section 4.5.4.

ing/updating interval and packets size, can be represented as:

$$C_o = \alpha \times K_{AVG} + \beta \qquad (4.5)$$

where α and β are constants while K_{AVG} is the average overlay node degree. In order to reduce the overhead introduced by overlay networks, we set $K_{AVG} \leq K_{MAX}$. In later analysis, we will discuss the impact of the overlay node degree value on the overlay performance in order to see if a trade-off value of overlay node degree can be found or not in order to achieve best performance whilst keeping a comparatively low overlay node degree.

4.3.2 Algorithm I: Least-Overlap MApping of Regular Graph (LO-MARG)

The overlay construction process can be formulated as a variation of the Bi-QAP problem and it is of high complexity to find the optimal solution in cases where the network size is larger than 20 [73]. As suggested by [74], a heuristic method exploiting Simulated Annealing (SA) is proposed to solve this problem.

The basic flow of the Least-Overlap Mapping algorithm is illustrated in Table 4.2. As indicated by the name, there is an additional constraint on the overlay structure. We assume that the overlay nodes all have the same node degree, i.e. the overlay topology is described by a regular graph. One of the reasons we consider a regular graph is because it has been reported to have good performance in designing optimal network topologies under arbitrary constraints by other researchers [75, 76]. Moreover, a regular graph can withstand a higher number of node/link failures than other graphs with node degree not all equal to K_{MAX} (i.e. some nodes have a degree smaller than K_{MAX} with the rest equal to K_{MAX}). Last but not least, there is no selection preference[3] in terms of underlying node connectivity and our aim is to maintain as high a connectivity as possible in the

---

[3]In network-provider-independent overlays, the nodes with higher connectivity in the underlying layer are preferred and thus termed as "selection preference" here.

Table 4.2: The LO-MARG algorithm

Initialization:	
(1) $T(0) = T_{initial}$, $G_O(N_O, E_O)$ is assigned as a regular graph $G_{reg}^{K_{MAX}}$. (2) $\{M_j^I, 0 \leq I <\mid N_I \mid\}$, randomly map overlay nodes to underlying nodes as current solution. (3) Calculate $Cost(V(0)) = \sum_I \sum_{J, J \neq I} OL(L_I, L_J)$.	
Procedure	
Step 1	
Step 1.1	$T = T(t)$; Find a neighbouring solution by randomly exchanging two selected overlay nodes (which still results in a regular graph), $V_{temp}(t) = f_{Ij,Km}(V(t))$;
Step 1.2	Calculate the cost difference between the two solutions and $\Delta Cost = Cost(V_{temp}(t)) - Cost(V(t))$;
Step 1.3	If $\Delta Cost < 0$, then accept $V_{temp}(t)$ and $V(t+1) = V_{temp}(t)$; otherwise, calculate the probability of accepting $V_{temp}(t)$, $Pr = exp(-\Delta Cost/T)$. If $Pr > \lambda$, accept $V_{temp}(t)$ and $V(t+1) = V_{temp}(t)$; otherwise, discard $V_{temp}(t)$, where λ is randomly generated number and $0 < \lambda < 1$.
Step 1.4	If the iteration number at current temperature is smaller than N_{try}, go to **Step 1**; otherwise go to **Step 2**.
Step 2	
Step 2.1	$T(t+1) = T(t) \times \mu$, $t = t + 1$
Step 2.2	If the iteration number of cooling process, **aka Step 2.1**, is smaller than N_{max}, go to **Step 1**; otherwise, the procedure terminates.

overlay layer. A regular graph is thus more desirable than other graphs under the same constraint. Although there are discussions of how to find regular graphs with different features from the random graph family [77], we assume the regular graph is fixed in later discussions for simplicity [4].

4.3.3 Algorithm II: Enhanced Dual-Layer-aware KMST (EDL-KMST)

Zhi Li *et al* [48] proposed a heuristic solution to the overlay topology construction problem in the context of provider-independent overlays. The basic idea is to minimize

---

[4]We have discussed the impact of a regular graph with different girths and it is presented in Appendix B.

the overlapping of underlying links between two consecutive Minimum Spanning Trees (MST) that are constructed. Based on the conjecture that there is a higher possibility of overlap among the virtual links in the provider-independent overlays, we adapt and enhance their algorithm to be implemented in our network scenarios in order to minimize the overlap for the overlay as a whole.

"Dual-layer-awareness" means not only the overlay network constraints but also underlying network topological information is considered during the topology selection process. Instead of updating the weight of virtual links according to the overlap number in term of substrate hops between the MSTs, the proposed algorithm checks the overlap and updates the weights during the MST construction process. Thus, it can ensure the least overlap of the newly selected virtual links with the previous selected set of overlay links.

The EDL-KMST algorithm shares the same objective with the LO-MARG algorithm in minimizing the overlap of overlay links. However, different from the LO-MARG algorithm, the proposed dual-layer-aware KMST scheme cannot guarantee the degree of all the overlay nodes will be identical. Nevertheless, it is less complex to complete as it incrementally selects a subset of virtual links. The basic flow of this algorithm is described in Table 4.3.

4.4   Overlay Node Selection Process

In this chapter, we assume the node set for the overlay is randomly chosen from all the eligible substrate nodes and fixed for the performance evaluation. However, in order to avoid scenarios where two randomly chosen overlay nodes share a same upstream node as discussed in Chapter 3, we employ the random selection process depicted in Figure 4.2 during our simulation.

Table 4.3: The EDL-KMST algorithm

Step 1:	Construct a temporary full mesh overlay topology $G_O^{FM}(N_O', E_O')$, in which $\{\theta(I, J) = 1, 0 \leq I, J < \mid N_O \mid\}$ and W_{L_I} is initiated as the number of substrate nodes each virtual link goes through;
Step 2:	Find a minimum spanning tree connecting all the overlay nodes subject to $K_I \leq K_{MAX}$ by recursively adding a virtual link in the output overlay topology $G_O(N_O, E_O)$ and update the weight of the links in $G_O^{FM}(N_O', E_O')$ using the following rules: (1) For the newly chosen overlay link, set its weight to maximal value thus it cannot be selected in subsequent MST construction process; (2) For an unselected virtual link, if it shares x number of substrate nodes with the selected virtual link, update its weight as $W_{L_I} = W_{L_I} + x$.
Step 3:	Repeat **Step 2** until each overlay node meets the overlay degree constraint K_{MAX} or no additional link can be added into the overlay topology.

4.5 Performance Evaluation with AS-level Topologies

4.5.1 Assumptions

During the simulation process, the following are assumed to be true:

1. The virtual link between two adjacent overlay nodes follows a series of AS hops that are determined using a least cost routing algorithm. Only symmetric routes are considered for simplicity.

2. It is assumed that the substrate topology is known to the overlay through active probing methods. AS topological information can be obtained by exploiting publicly available IP-to-AS mapping tools [78].

3. All the overlay nodes are assumed to have the ability to detect the performance
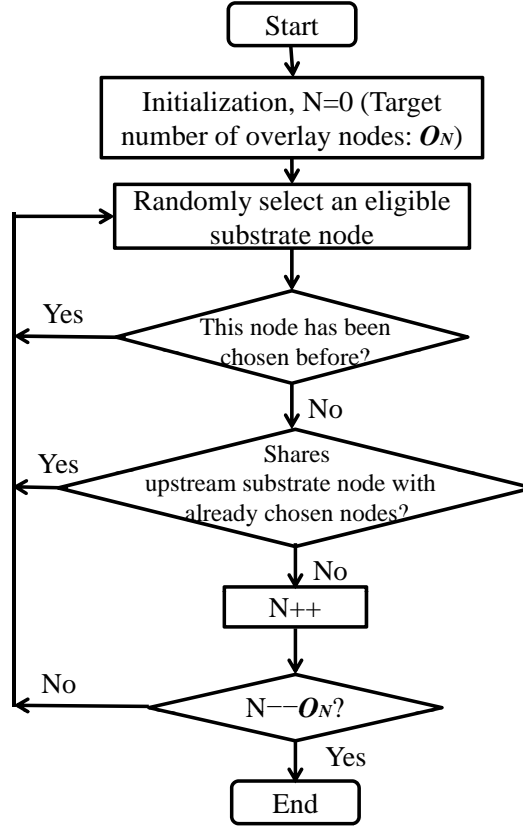
Figure 4.2: The random overlay node selection process

degradation and/or the failures of the substrate network in a timely manner. They get up-to-date routing information based on link-state routing that operates across the overlay at the time of making routing decisions.

4.5.2 AS-level Topologies

We investigate scenarios using the AS-level topologies as the substrate topology for several reasons: (1) the overlay node candidates are assumed to reside in the ASes with lower connectivity, therefore it is reasonable to include one overlay node in each domain as they tend to have the same outgoing paths to the same destinations; (2) inter-AS problems such as the BGP re-convergence time is comparatively much longer than that within an AS and affects the AS as a whole; (3) there is a reported characteristic of the AS-level topology, namely the power-law feature, which mimics that of the Internet.

We have exploited the Orbis topology generator with the Skitter [36] and Whois [37] datasets as inputs. Since the Whois dataset has been reported to have different features to that of Skitter [79], such as better connections between nodes with medium degrees, it is conjectured that the performance gap between various topology construction methods might differ. Although Orbis can generate re-scaled topologies retaining different topological properties, we only focus on re-scaled topologies that can retain the joint node degree distribution according to the analysis presented in [79]. We use Skitter-based and Whois-based to denote these two re-scaled topologies generated with network size around 2000 nodes.

4.5.3 Failure Models and Evaluation Metric

In this section, we only focus on the resilience of various constructed overlay topologies under AS-node failures. Furthermore, a "failure" does not necessarily mean a physical breakdown but can also represent performance degradation (e.g. delay or loss) below an acceptable threshold as perceived by the overlay monitoring process.

Three failure models are implemented for performance analysis. They are the Random Single Failure model, the Random Multiple Failure model and the Accumulative Focused Failure model, respectively. The last one is similar to large-scale failure scenarios in [80][5]. The first two are self-explanatory. With the third approach, the failure starts from a single AS and propagates to its neighbours. All the neighbours of the previous failure set will be viewed as failed by turns.

For the resilience evaluation of the overlay network, failure of ASes that do not support overlay virtual links are not considered. Therefore in the simulations, the failures are randomly selected from the subset of ASes that the overlay network virtual links traverse. We therefore introduce the term "ON Supporting AS Failures" to represent the number of failed ASes that are covered by the Overlay Network (ON). However,

---

[5]Geographical information is not considered herein since we do not have access to such information.

for the third failure model, the radiating effect will include both covered and uncovered substrate nodes, so the first failure is chosen from the whole substrate node set.

The metrics we use include the resilience of the topologies constructed using various algorithms and the relative resilience ρ_r which is defined as the ratio of the resilience obtained by the overlay constructed with an algorithm compared to that of a full mesh overlay topology. The latter one is formally defined as follows:

$$\rho_r = \frac{\rho}{\rho*} \tag{4.6}$$

where the resilience $\rho*$ of a full mesh overlay topology is the best that can be obtained under different failure models. Therefore, the closer ρ_r is to 1, the better the performance of the constructed overlay is to that of the optimal resilience performance.

4.5.4 Comparison Methods

Besides the two proposed algorithms explained in Section 4.3, we also propose a simple algorithm called Random Mapping of Regular Graph (termed RM-RG). This method tries to maintain regularity in the overlay topology but has no intention of minimizing the overlap when mapping the overlay topology onto the underlying layer.

There are four other existing algorithms implemented here for comparison purposes and are listed as follows:

1. **Full Mesh (FM)**: This topology provides the best resilience possible whilst incurring the largest overhead according to equation (4.5). As a full mesh topology does not obey the overlay node degree constraint, it is included here only as a benchmark for the resilience performance evaluation;

2. **Topological-aware K Random Connection (TKRC)**: This method [48] builds a Minimum Spanning Tree (MST) first and then selects least overlapping virtual links for each node in turn whilst limiting the overlay node degree;

3. **Topological-aware K joint-Minimum Spanning Tree (TKMST)**: This method [48] builds K minimum spanning trees in turn and makes sure the tree selected will overlap the least with previous trees whilst limiting the overlay node degree;

4. **K Random Connection (KRC)**: This method only maintains the average of the overlay node degree equal to that of LO-MARG algorithm for comparison purposes.

4.5.5   Results and Analysis

This section is devoted to exploring the performance of the three proposed algorithms with different settings given the two AS-level topologies. There are some parameters that are configured the same throughout performance evaluation: (1) the configuration employed and the cost function for the simulated-annealing-based LO-MARG algorithm are discussed and specified in Appendix A; (2) the resilience evaluation metric is averaged over 1000 iterations and the justification for this setting is presented in Appendix B. The substrate node candidates are assumed to reside in stub ASes for the performance evaluation unless otherwise stated. As for the verification of the simulator correctness and discussion of some typical settings/assumptions, please refer to Appendix B for details.

We present our analysis together with results. In (1), we investigate the impact of overlay node degree on overlay topologies constructed using various algorithms. In (2), we discuss the impact of different numbers of overlay nodes. In (1) and (2) we use the Random Multiple Failure Model and in (3) we present the results with Single Failure Model and Accumulative Focused Failure Model.

(1) We investigate the impact of overlay node degree with both Skitter-based and Whois-based AS-level topologies. Figure 4.3 depicts the resilience of overlay topologies with different overlay node degrees constructed using the LO-MARG algorithm against multiple failures with the Skitter-based topology serving as the substrate network. As shown in the figure, the higher the overlay node degree is, the higher the
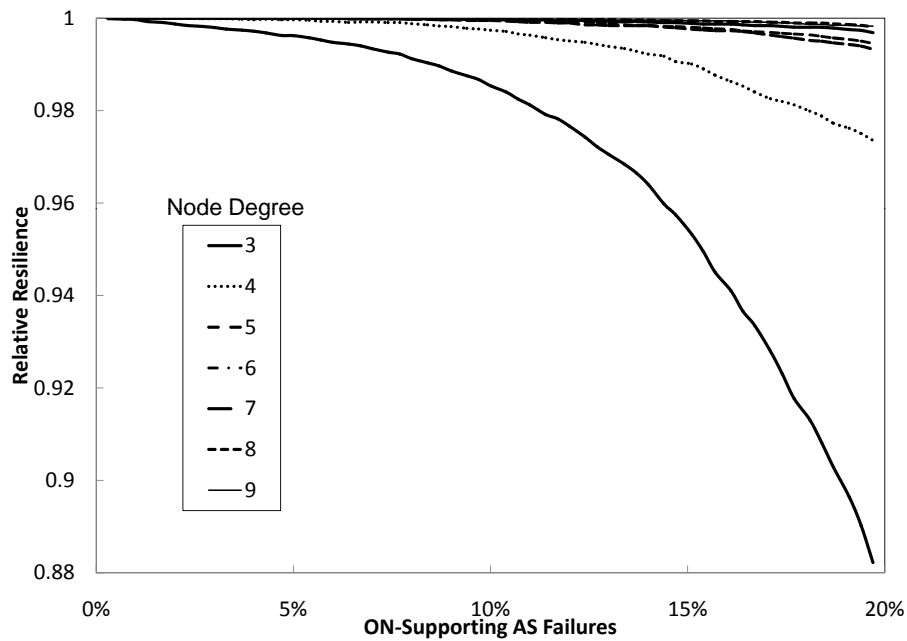
Figure 4.3: Discussion of the overlay node degree setting with 50 overlay nodes in the Skitter-based AS-level topology

network resilience is and closer it is to that of the full mesh topology. As it has already been shown in Equation (4.5) the higher overlay node degree is, the higher is the overhead cost. Therefore, an overlay with a full mesh topology introduces the highest amount of overhead whilst maintaining the best resilience possible. However, as the overlay node degree increases, there is a diminishing increase in terms of the resilience improvement. Therefore, it is reasonable to select a comparatively low overlay node degree in order to achieve an acceptable trade off between the overhead and performance improvement gained herein.

As shown in Figure 4.4, the proposed LO-MARG algorithm performs much better as compared to other methods when the overlay node degree is low. Substrate-topology-aware algorithms including the proposed EDL-KMST and the existing TKMST and TKRC algorithms perform better than ones oblivious to the substrate topology and do not follow the regularity in the overlay node degree. Surprisingly, the RM-RG algorithm performs comparably well to that of the TKMST algorithm although it does not take into consideration substrate information but only uses a regular graph. As the node
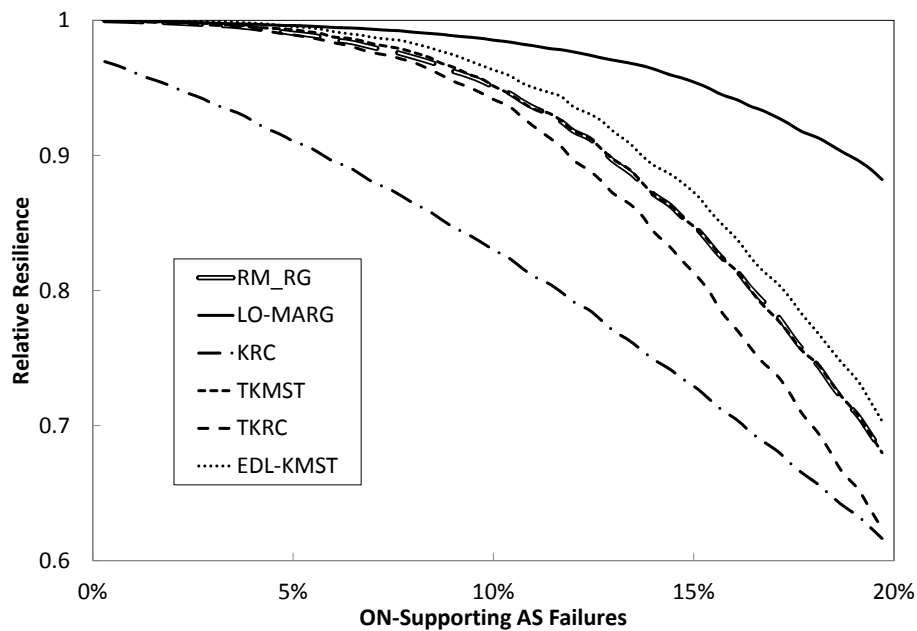
Figure 4.4: The relative resilience performance of various algorithms with overlay node degree equal to 3 and the number of overlay nodes equal to 50 with the Skitter-based AS-level topology

degree increases, the performance difference among different methods becomes smaller. As shown in Figure 4.5, the performance of the RM-RG algorithm is very close to that of the substrate-topology-aware algorithms.

Figure 4.6, Figure 4.7 and Figure 4.8 provide similar results but for the Whois-based topology. Although similar observations are obtained, unlike the evaluation with the Skitter-based topology, the performance gap between the RM-RG algorithm and the substrate-topology-aware algorithms is bigger. This supports our hypothesis that the joint connectivity of nodes with mid-range degrees will have an effect on the overlay resilience performance. Nevertheless, the resilience of the RM-RG algorithm is still relative small and close to that of the full mesh given an appropriate overlay node degree. For our later discussion, we fix the overlay node degree equal to 5 since the resilience improvement with higher overlay node degrees is small.

(2) We change the number of the overlay nodes to verify whether the previous obser-
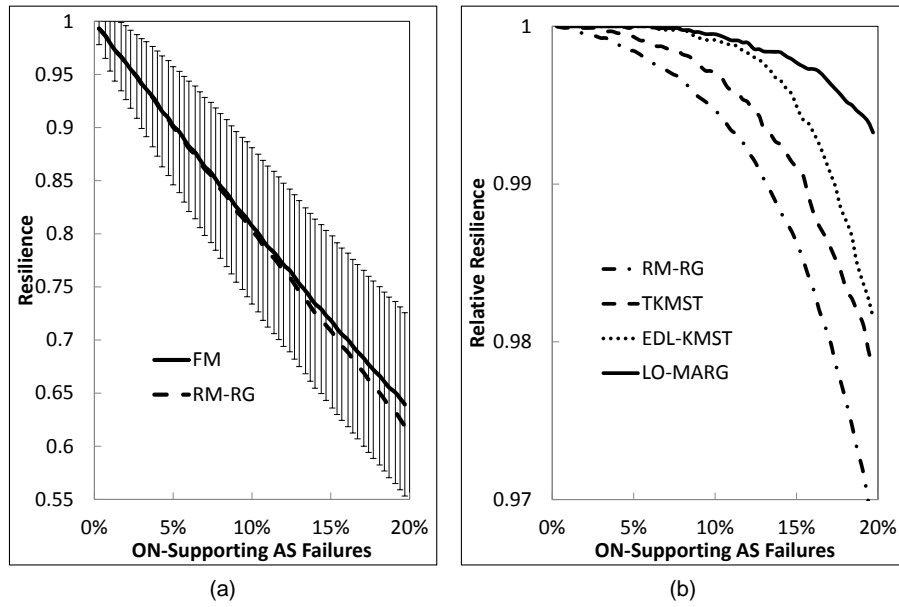
Figure 4.5: Results of various algorithms with overlay node degree equal to 5 and the number of overlay nodes equal to 50 with the Skitter-based AS-level topology: (a) the resilience performance of Full Mesh with one standard deviation and the RM-RG algorithm; (b) the relative resilience of the LO-MARG, TKMST, EDL-KMST and RM-RG algorithms
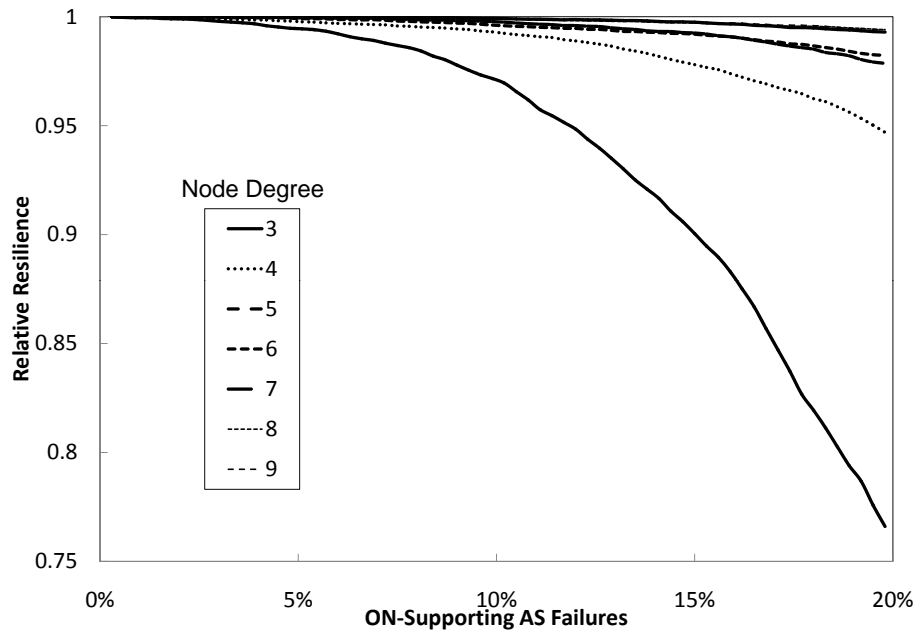
Figure 4.6: Discussion of the overlay node degree setting with 50 overlay nodes with the Whois-based AS-level topology
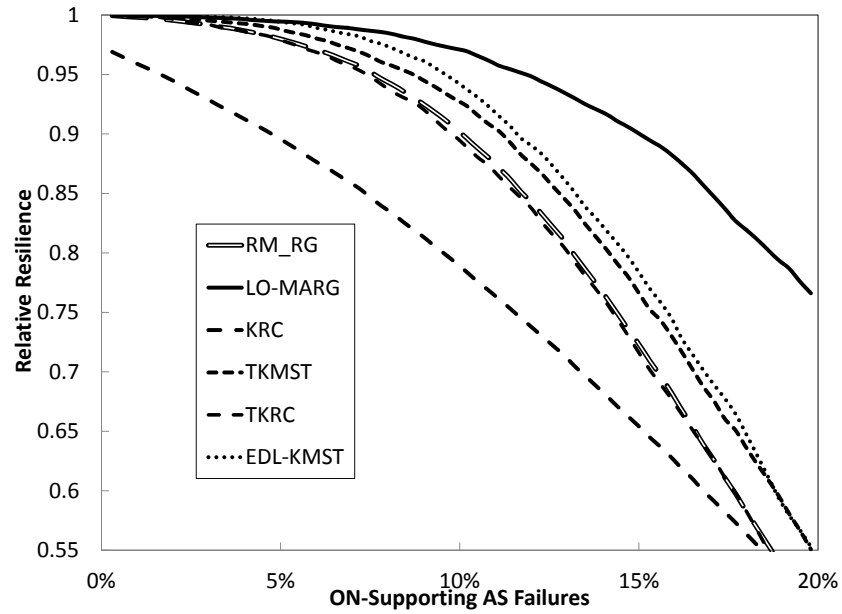
Figure 4.7: The relative resilience performance of various algorithms with overlay node degree equal to 3 and the number of overlay nodes equal to 50 with the Whois-based AS-level topology
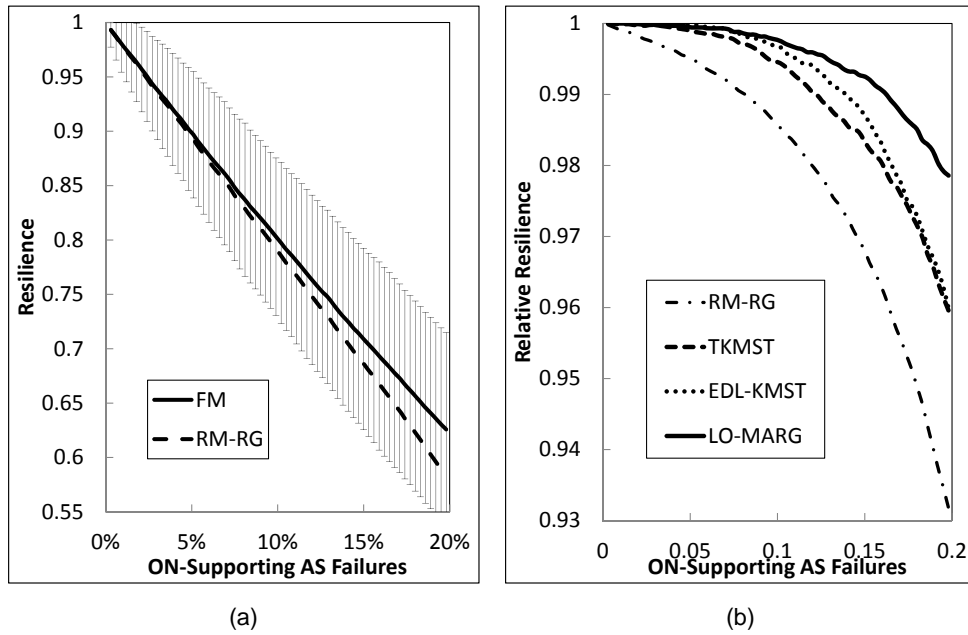
Figure 4.8: Results of various algorithms with overlay node degree equal to 5 and the number of overlay nodes equal to 50 with the Whois-based AS-level topology: (a) the resilience performance of Full Mesh with one standard deviation and the RM-RG algorithm; (b) the relative resilience of the LO-MARG, TKMST, EDL-KMST and RM-RG algorithms
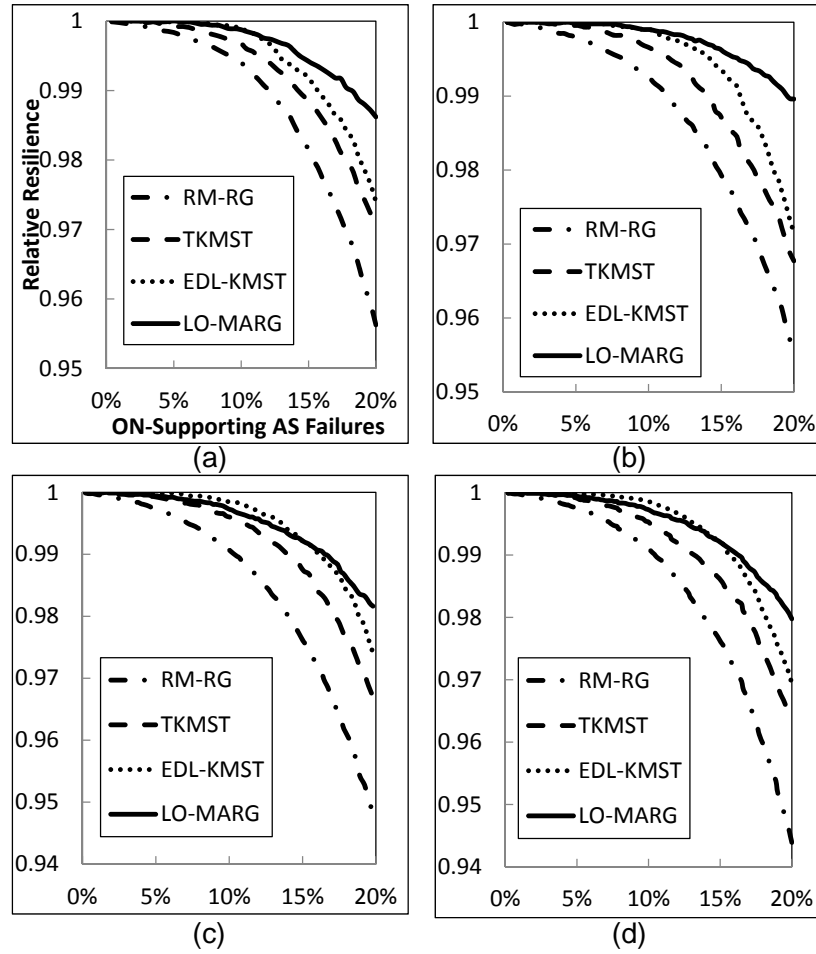
Figure 4.9: The relative resilience performance with different overlay sizes with the Skitter-based AS-level topology: (a) 40; (b) 60; (c) 80; (d) 100

vations still hold for different scenarios. The results are shown in Figure 4.9 [6] and Figure 4.10 for the Skitter-based and Whois-based AS-level topologies, respectively. As shown in the graphs, the results still support our previous analysis.

(3) The results shown in Figure 4.11 and Figure 4.12 illustrate when the single failure and accumulative focused failure models are deployed. It can be concluded that there is little difference between various methods in both models. The resilience of the topologies

---

[6]The LO-MARG algorithm is not the best when the overlay node number is 80 and 100 nodes in this figure due to the setting of the simulated annealing parameter settings. Since the difference between the best one (EDL-KMST) and the LO-MARG algorithm is within 0.2%, we choose not to go through the process presented in Appendix A again. However, it can be undertaken by following the parameter setting process described in Appendix A in order to achieve better performance with the LO-MARG algorithm.
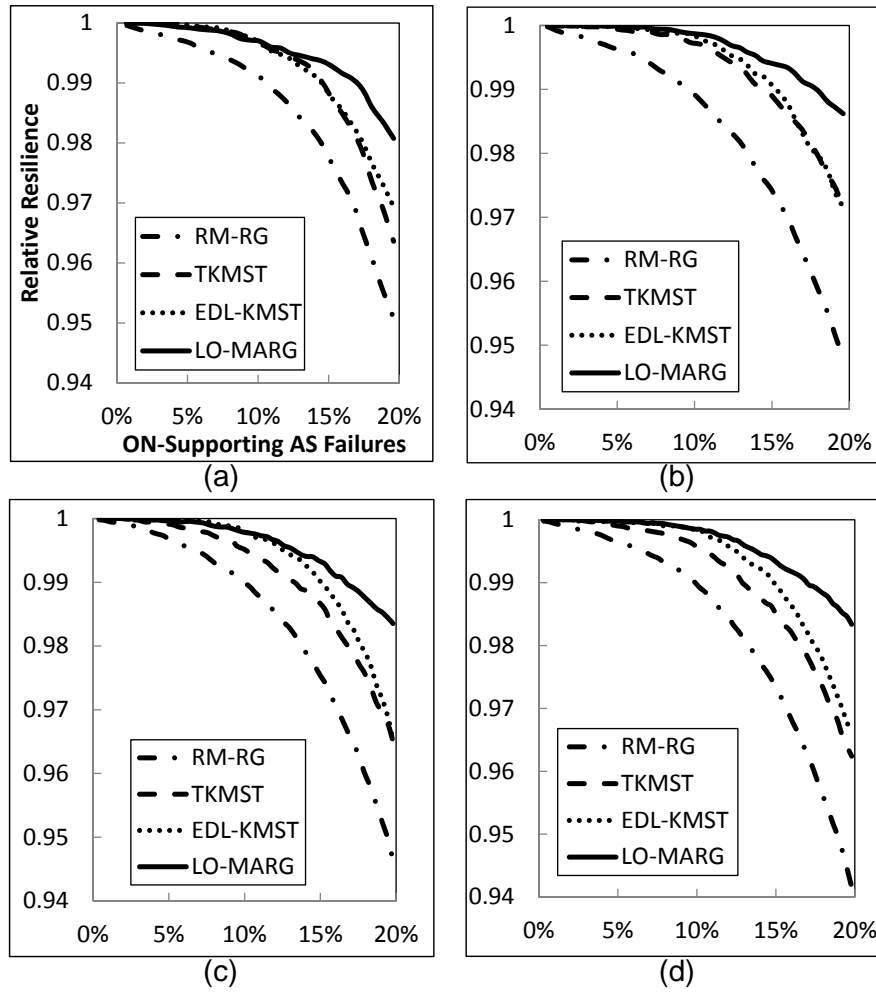
Figure 4.10: The relative resilience performance with different overlay sizes with the Whois-based AS-level topology: (a) 20; (b) 30; (c) 40; (d) 50

constructed using various methods are high against a single failure. This is because the overlay networks have high remaining connectivities after one random single failure. Thus, the failure can be easily recovered by the overlay network although it cannot be mitigated quickly using the traditional paradigm. As for the accumulative focused failures, these results originate from the fact that it has devastating effect on overlays as it can affect the all the neighbouring substrate nodes starting from the failure starting point. Thus, irrespective of the methods employed, the connection loss in the overlay layer is very high.
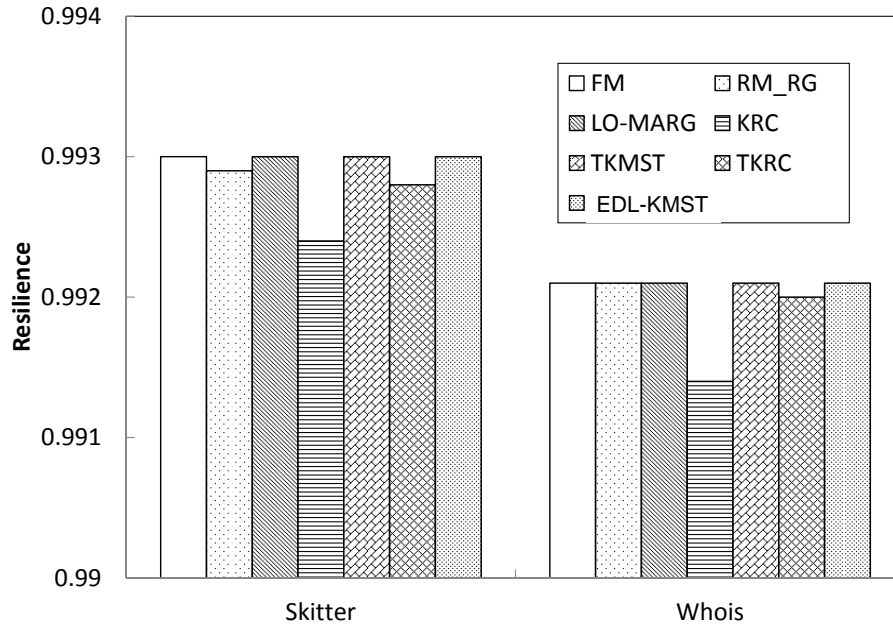
Figure 4.11: Impact of single AS-node failures in both Skitter-based and Whois-based topologies
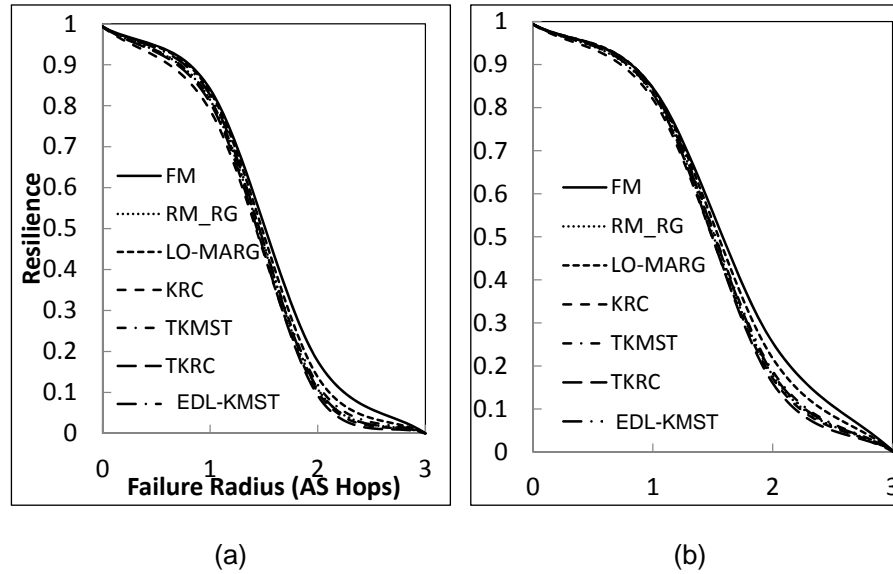
Figure 4.12: Impact of the cumulative focused failure model with both Skitter-based and Whois-based topologies (The x-axis represents the Failure Radius of the accumulative-focused failure model. Specifically, 0 represents the failure of a single AS and x (i.e. 1-3) means all the nodes that are x AS hops away from this point are deemed to have malfunctioned, too.)

4.6 Performance Evaluation with Router-level Topologies

4.6.1 Assumptions and Simulation Settings

The assumptions are similar to those for the evaluation with AS-level topologies described in Section 4.5 except that we do not assume a symmetrical path between a substrate node pair and it is determined by the shortest path first algorithm implemented in the substrate layer.

As for the router-level topology formed among the ROMCA overlay nodes, this information can be inferred by the techniques summarized in Chapter 6. The GT-ITM topology generator is exploited to generate different transit-stub topologies. We adopt the same failure model as used in [48], namely setting the substrate link failure magnitude to be 2% and the results are averaged over 1000 iterations unless otherwise stated. Accordingly, the TKMST and EDL-KMST algorithms use link overlap as the metric when constructing the overlay links sequentially.

Due to the high complexity of the LO-MARG algorithms and the good performance of the EDL-KMST and RM-RG algorithms given appropriate settings, both of which have comparatively very low complexity, we mainly focus on the evaluation of the EDL-KMST and RM-RG algorithms.

4.6.2 Results and Analysis

Firstly, we discuss the impact of the overlay topology degree with one of the generated substrate router-level topologies. As shown in Figure 4.13, the EDL-KMST algorithm performs the best among all the methods with the same overlay node degree given various overlay node degrees. But there is only a small difference between the EDL-KMST algorithm and the TKMST algorithm. Furthermore, the higher the overlay node degree is, the higher the network resilience is and closer to that of the full mesh topology.
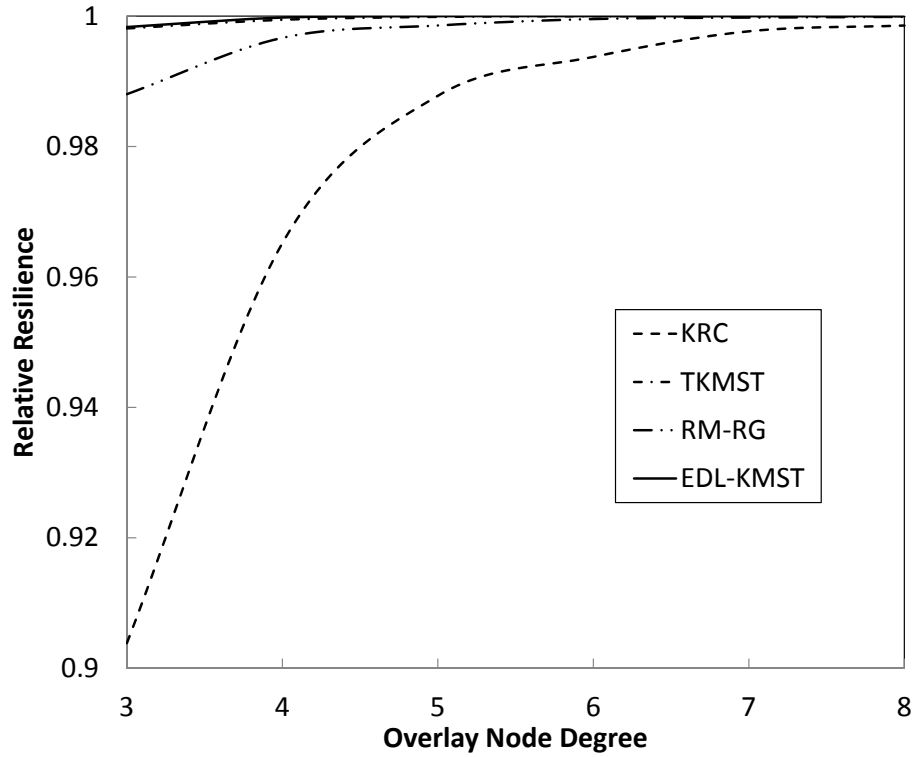
Figure 4.13: Impact of overlay node degree in a router-level topology with 3200 nodes and about 20000 links

In order to achieve comparable performance with that of the full mesh topology with much less overhead, we choose to set the overlay node degree to 4 where the RM-RG algorithm can perform satisfactorily.

Secondly, we have also verified the performance of the EDL-KMST and RM-RG algorithms with different overlay node numbers and different substrate networks. The results are shown in Figure 4.14 and Figure 4.15. As shown in both graphs, the performance of the EDL-KMST algorithm is best. Moreover, given the current settings of overlay node degree, the performance of the RM-RG algorithm has comparable performance to that of the substrate-topology aware algorithms.
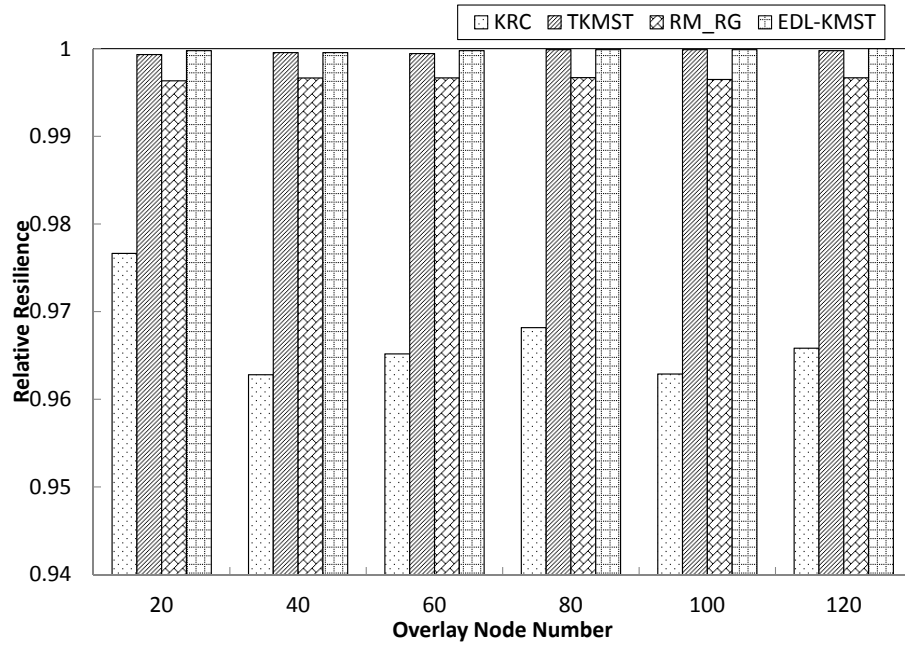
Figure 4.14: Impact of overlay node number in a router-level topology with 3200 nodes and about 20000 links (60 overlay nodes with overlay node degree equal to 4)
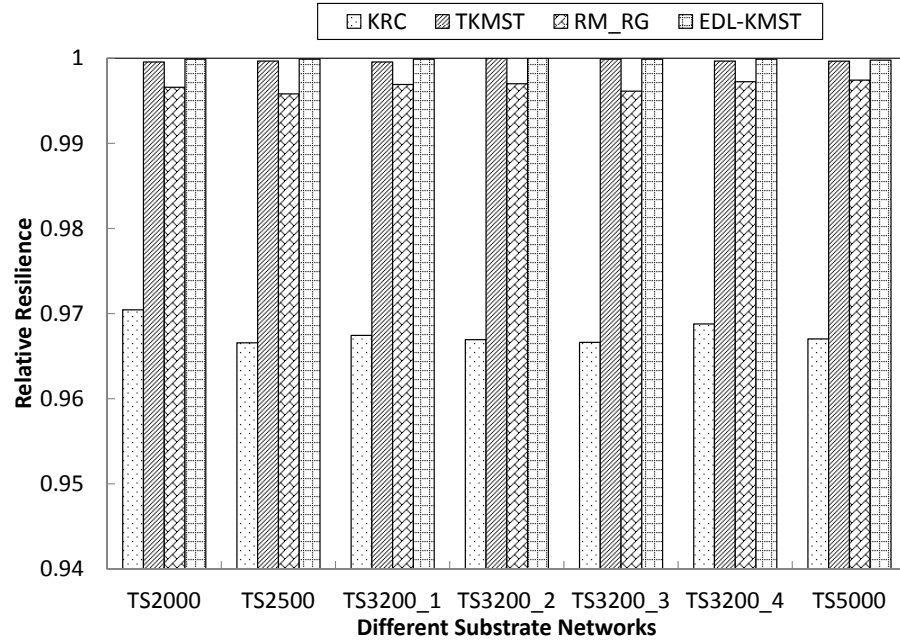
Figure 4.15: Results with different substrate networks with 60 overlay nodes and overlay node degree equal to 4 (Note: in the x-axis, the four ts3200 topologies have the same number of nodes but generated with different random number seeds in the GT-ITM topology generator.)

4.7   Summary

In this chapter, the overlay topology construction issue in the context of the ROMCA overlay for providing resilience service is thoroughly analyzed using both AS-level and router-level topologies as the substrate network. Several conclusions can be identified and are listed as follows:

- The proposed Least-Overlap Mapping of Regular Graph algorithm (LO-MARG) performs the best in all scenarios with AS-level topologies and it performs much better than other methods given a low overlay degree constraint. However, it possesses very high computational complexity since a large number of iterations are needed to search for a more resilient solution.

- The proposed Enhanced Dual-Layer-aware K-Minimum Spanning Tree algorithm (EDL-KMST) typically performs second best in terms of resilience among all the methods compared under the same overlay node degree constraint with AS-level topologies and performs the best with router-level topologies compared to the methods considered. This EDL-KMST algorithm has much lower complexity; however, it cannot guarantee that the constructed overlay topology is a regular graph.

- Through extensive simulations with both AS-level and router-level topologies, it can be concluded that a random mapping of a regular graph method can perform satisfactorily with an appropriate overlay node degree setting. The advantage of this method is that it does not need substrate topology information. However, if an overlay with guaranteed higher resilience is desirable, the two proposed substrate-topology-aware schemes are recommended.

Chapter 5

Application Mapping to Achieve Enhanced Resilience and QoS

5.1 Overview

In this chapter, we investigate how to map an application request exploiting the ROMCA overlay in such a way that the selected hosting overlay node and link sets can provide not only enhanced QoS but also resilience against potential substrate link failure(s). Unlike previous works which only strive to solve this problem heuristically, we first formulate it as an Integer Linear Program (ILP). Then, we enhance the proposed ILP model taking into consideration the substrate topology features so as to provide effective backup paths. Furthermore, in order to solve the problem time-efficiently, a novel and effective heuristic which considers the substrate network topology information is also proposed for use with larger networks. The effectiveness of both the proposed ILP model and the new heuristic is verified through extensive simulations. By using small synthetic topologies, the proposed enhanced ILP model is proved to perform significantly better in terms of QoS performance as compared to the best existing heuristic and the heuristic solution proposed in this chapter. Moreover, the ILP model can provide diversified working and

backup paths using only few additional overlay nodes. However, the ILP does not scale well. The proposed heuristic can enforce substrate-diversified paths but with a higher number of additional overlay nodes involved in the mapping solution.

5.2 Problem Description

Motivated by network virtualization aimed at improving the effectiveness of the Internet and the desire to support diversified services, deploying an application onto a host network in a flexible and effective way has attracted much attention [15, 16]. Application mapping is defined here as the process of mapping an application request onto a host network (in our case, the ROMCA overlay) whilst satisfying the specific requirements of the application. Typical requirements include the capacity of the nodes and links in the host network, location of the host nodes, and the QoS performance and topology constraints of the application request.

Depending on the components that need to be mapped, application mapping can be categorized into two kinds: (1) link mapping of an application request, assuming that nodes supporting/accommodating the application are fixed; (2) both node and link mapping of an application request. As an example of the first category, S. Roy *et al* [43] explore how to optimize an application-specific performance metric by selecting a subset of substrate candidate nodes. The selected nodes are used to provide alternate paths for an application and this can improve the performance such as resilience and throughput. A typical example of the second category is Virtual Network Assignment (VNA) [81–84], where both the nodes and links of the virtual network need to be mapped onto a substrate network. The common requirement associated with this example is the capacity of nodes and links in the hosting network [15]. Another example of the second type is the work carried out in [16, 19], aimed at finding a set of links and nodes such that the QoS requirement can be enhanced together with the resilience guarantees. Such applications usually involve a distributed cooperative environment, such as distributed

gaming, simulation and high performance computing. In this chapter, we focus on the second kind of application mapping where both links and nodes are considered when performing the mapping and the host network is the ROMCA overlay.

Besides the QoS requirements of an application request, one of the key challenges in application mapping is how to increase the resilience of the mapped application. This problem can be stated as: *given a host network with QoS and candidate node and link information, how should one place an application request with QoS constraints so as to maximize the performance of the mapped application with the least resources possible of the host network?* The problem is illustrated in Figure 5.1. Similar to the work in [16, 85], we mainly focus on meeting the delay requirement of an application request in this chapter. Other QoS metrics, such as loss rate, can be easily considered by adding additional constraints to the proposed ILP model. The candidate nodes for hosting an application request are assumed to be the overlay nodes present in the ROMCA overlay. When designing the model, we consider the following two objectives for fulfilling an application request exploiting the ROMCA overlay, namely:

1. Finding a mapping of the application onto the host overlay layer in such a way
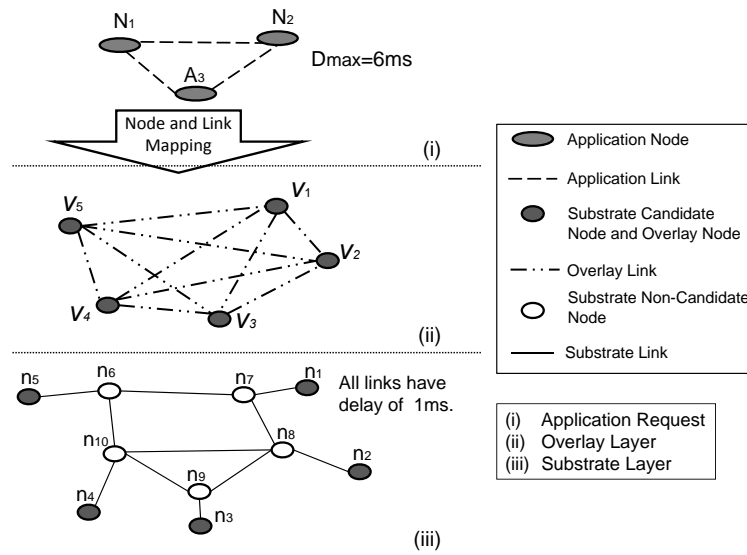
Figure 5.1: Problem description

that the resulting delay performance is best.

2. Providing resilience by finding backup paths for the application request connections in the mapping solution. However, this may require additional overlay nodes to be involved in the mapping configuration thus incurring additional cost. Hence, constraints on additional overlay node resource usage are considered in our model.

5.3   Integer Linear Program Formulation

In this section, we formulate the above-mentioned problem as an Integer Linear Program. Furthermore, we strengthen the basic ILP model by incorporating substrate topology information which can be obtained from the ROMCA overlay through active topology inference. All the inputs to the models are defined in Table 5.1 whereas the variables are listed in Table 5.2.

Table 5.1: Input (constant) notation table

Notation	Definition
Notations for the application request and the substrate network	
$\{N_I\}, \{L_{AB}\}$	Set of nodes and links that describe an application request
D_{max}	The delay constraint set by the application request
$\{n_k\}, \{l_{ij}\}$	Set of nodes and links that describe the substrate network
$\{p_{ij}\}$	The path between two nodes n_i and n_j in the substrate network layer
D_{ij}	Delay of a path p_{ij} in the substrate network
Notations for the ROMCA overlay	
$\{\nu_{r'}\}, \{\lambda_{m'n'}\}$	Set of overlay nodes and QoS-compliant overlay links to describe the simplified overlay topology
$ol_{m'n',p'q'}$	The overlap between two overlay links $\lambda_{m'n'}$ and $\lambda_{p'q'}$ in the simplified overlay topology. It can be measured by the number of substrate nodes/links two overlay links share in common
$D_{m'n'}$	Delay of an overlay link $\lambda_{m'n'}$ in the simplified overlay topology
Notations for the ILP Model	
α, β, γ	Weight factors (constants)

Table 5.2: Variable notation table

Notation	Definition
$W_{m'n'}^{AB}$	(Binary)=1 if an application link L_{AB} is mapped on $\lambda_{m'n'}$, otherwise 0.
$B_{m'n'}^{AB}$	(Binary)=1 if an overlay link $\lambda_{m'n'}$ is used as the backup path for an application link L_{AB}, otherwise 0.
$M_{r'}^{I}$	(Binary)=1 if an application node N_I is hosted by a candidate overlay node $\nu_{r'}$, otherwise 0.
$\Phi_{m'n',p'q'}$	(Binary)=1 if two overlay links $\lambda_{m'n'}$ and $\lambda_{p'q'}$ are both selected for application mapping, otherwise 0.
$B_{m'}$	(Binary)=1 if an overlay node $\nu_{m'}$ is used only for backup use in application mapping, otherwise 0.

5.3.1 Integer Linear Program Model

As described in Section 5.2, an application request has both the topology and QoS requirements. In order to facilitate the mapping of the application request and reduce the computational complexity of the ILP model, we thus construct a simplified overlay topology $(\nu_{r'}, \lambda_{m'n'})$.

The node set in this simplified topology consists of the candidate overlay nodes for application mapping. The link set in this simplified overlay topology is composed of the QoS-compliant overlay links. Each of these links is actually a path in the substrate layer. Thus, the delay of an overlay link is the sum of the delay of each link along the related path in the substrate layer (i.e. $D_{m'n'} = D_{ij}$). QoS-compliance denotes that the delay of each overlay link is no bigger than that required by the application, namely,

$$\{\lambda_{m'n'}\} = \{p_{ij} \in \{p_{lk}\} \mid n_i = \nu_{m'},$$

$$n_j = \nu_{n'}, D_{ij} \leq D_{max}\} \tag{5.1}$$

Figure 5.1 provides an example of the formation of a simplified overlay topology where there are five nodes in the overlay network. According to the delay requirement of the application request depicted and assuming shortest-path-first routing in the substrate layer, the overlay links between these candidate overlay nodes form a full mesh topology
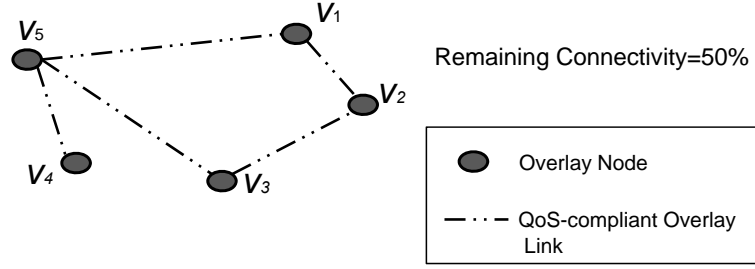
Figure 5.2: A simplified topology with 50% remaining connectivity

as shown in Figure 5.1 (ii). The ratio of the number of links in the simplified topology as compared to that of the full mesh topology is defined as the remaining connectivity of the simplified topology. An example of a simplified topology with 50% remaining connectivity is depicted in Figure 5.2.

If the substrate topology information is not considered, as in [19], the ILP formulation can be presented as follows.

Objective:

$$minimize \quad \alpha \sum_{\{L_{AB}\}} \sum_{\{\lambda_{m'n'}\}} D_{m'n'} W^{AB}_{m'n'} + \beta \sum_{\{\nu_{m'}\}} B_{m'} \tag{5.2}$$

Constraints:

-Delay Related Constraints:

$$\sum_{\{\lambda_{m'n'}\}} D_{m'n'} W^{AB}_{m'n'} < D_{max}, \quad \forall L_{AB} \tag{5.3}$$

$$\sum_{\{\lambda_{m'n'}\}} D_{m'n'} B^{AB}_{m'n'} < D_{max}, \quad \forall L_{AB} \tag{5.4}$$

-Flow Related Constraints:

$$\sum_{m'} W_{m'n'}^{AB} - \sum_{p'} W_{n'p'}^{AB} = \begin{cases} -1 & \text{if } M_{n'}^A = 1 \\ 1 & \text{if } M_{n'}^B = 1 \quad \forall L_{AB} \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

$$\sum_{m'} B_{m'n'}^{AB} - \sum_{p'} B_{n'p'}^{AB} = \begin{cases} -1 & \text{if } M_{n'}^A = 1 \\ 1 & \text{if } M_{n'}^B = 1 \quad \forall L_{AB} \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

-Working and Backup Path Constraints:

$$W_{m'n'}^{AB} + B_{m'n'}^{AB} \leq 1 \quad \forall L_{AB}, \forall \lambda_{m'n'} \tag{5.7}$$

-Node Mapping Constraints:

$$\sum_{\{N_I\}} M_{r'}^I \leq 1, \quad \forall \nu_{r'} \tag{5.8}$$

$$\sum_{\{\nu_{r'}\}} M_{r'}^I = 1, \quad \forall N_I \tag{5.9}$$

Remarks:

1. The objective function (5.2) of this ILP model encompasses the two goals of application mapping. The first half of the equation aims to minimize the delay of the mapped application request. The second part endeavours to minimize the number of overlay nodes which are used ONLY as a backup when allocating one backup path for each application link. In the objective function, $\alpha + \beta = 1$.

2. Constraint sets (5.3) and (5.4) ensure the delay upper bound set by the application

request can be accommodated during the selection of both the working and backup
substrate paths.

3. Constraint set (5.7) makes sure that working and backup paths for each link of the
   application request do not use the same overlay resource.

4. Constraint set (5.8) ensures that an overlay node can host no more than one appli-
   cation node whilst constraint set (5.9) ensures that an application node can be
   mapped onto no more than one overlay node.

5.3.2 Enhanced Integer Linear Program Model

As discussed by previous researchers [43, 48], two seemingly disjoint overlay paths may
share a common substrate node/link and thus their failure probabilities are not indepen-
dent. Using Figure 5.1 as an example, there are many solutions available for the applica-
tion request shown in this graph. We discuss only three possible solutions here, namely,
$\{\nu_1, \nu_2, \nu_3\}$, $\{\nu_2, \nu_3, \nu_4\}$ and $\{\nu_3, \nu_4, \nu_5\}$. However, only the second solution can pro-
vide effective resilience since the other two solutions cannot provide link-disjoint backup
paths [1]. Therefore, in order to further provide effective resilience during the mapping
process, we enhance our basic ILP model described in the last section by modifying the
objective function and adding supplementary constraints.

In the substrate network layer, it is assumed that the routing between two nodes
is determined by an underlying network routing algorithm (be it either Shortest Path
First(SPF) or policy-based routing). In the enhanced model, two aspects are considered.
The first one is that the working and backup paths are chosen by the overlay in such a
manner that they avoid link overlap in the substrate layer. The second one is that we
intend to minimize the substrate link overlap of two application links when mapped to
the overlay layer so as to minimize the impact of substrate link failure(s) on the mapped

---

[1]Access substrate links of the overlay nodes are not counted since disjointedness cannot be achieved
with them in provider-independent overlays like ROMCA.

links of the application request. Note that this model can be utilized only when the substrate topology information is available and it is assumed that the ROMCA overlay can obtain the accurate substrate topology information in this chapter.

The enhanced ILP overlay mapping model is presented as follows.

Enhanced objective function:

$$
\begin{aligned}
minimize\ \alpha \sum_{\{L_{AB}\}} \sum_{\{\lambda_{m'n'}\}} D_{m'n'} W^{AB}_{m'n'} + \\
\beta \sum_{\{\nu_{m'}\}} B_{m'} + \gamma \sum_{\{\lambda_{m'n'}\}} \sum_{\{\lambda_{p'q'}\}} ol_{m'n',p'q'} \Phi_{m'n',p'q'}
\end{aligned}
\tag{5.10}
$$

Here $\alpha + \beta + \gamma = 1$ [2] and $\Phi_{m'n',p'q'}$ is a binary variable and is equal to 1 only when $\sum_{\{L_{AB}\}} W^{AB}_{m'n'} = \sum_{\{L_{AB}\}} W^{AB}_{p'q'} = 1$. In order to linearize the expression of this binary variable, we use the following inequalities to describe this relationship.

$$
\Phi_{m'n,p'q'} \geq \sum_{\{L_{AB}\}} W^{AB}_{m'n'} + \sum_{\{L_{AB}\}} W^{AB}_{p'q'} - 1
\tag{5.11}
$$
$$
\forall \lambda_{m'n'},\ \forall \lambda_{p'q'}
$$

$$
\Phi_{m'n,p'q'} \leq \sum_{\{L_{AB}\}} W^{AB}_{m'n'} \quad \forall \lambda_{m'n'},\ \forall \lambda_{p'q'}
\tag{5.12}
$$

$$
\Phi_{m'n,p'q'} \leq \sum_{\{L_{AB}\}} W^{AB}_{p'q'} \quad \forall \lambda_{m'n'},\ \forall \lambda_{p'q'}
\tag{5.13}
$$

Additional Constraints:

$$
W^{AB}_{m'n'} + B^{AB}_{p'q'} \leq 1 \quad \text{if } ol_{m'n',p'q'} \geq 1,
\tag{5.14}
$$
$$
\forall \lambda_{m'n'},\ \forall \lambda_{p'q'},\ \forall L_{AB}
$$

Additional remarks:

1. The modified objective function (5.10) includes the objective to maximize the diver-

---

sity between each pair of mapped application links. Thus, the chance that substrate link failures cause the disruption of the mapped application service for the ROMCA overlay is reduced. Although it can be argued that it is better to find a mapping of application links without any overlap, we formulate it as stated above because it is likely that there will be no solution when there is high overlap in the paths among the candidate overlay nodes. However, this model can be easily adapted to achieve non-overlapping application link mapping by adding additional constraints instead of including the last part of the objective.

2. The additional constraint set (5.14) ensures that working and backup paths for an application link do not share underlying links wherever possible. Note that if a hosting overlay node is a stub node, the substrate link connecting this node with the rest of the overlay will be a single point of failure. For example, in Figure 5.1, all the overlay nodes are stub nodes. Hence, if their access link(s) fail, they will lose all of their connections without the possibility of finding an alternative. The access link overlap in the overlay layer is not taken into consideration when calculating the link overlap of two overlay paths among the candidate overlay nodes.

5.4 Proposed Heuristic Algorithm

Based on the previous heuristic proposed in [16], we develop a new and simple heuristic that incorporates substrate topology information. This substrate-topology-aware heuristic for finding an application mapping solution ensures that the allocated backup paths do not share common links in the substrate layer with their corresponding working paths. The procedure for this heuristic is illustrated in Table 5.3.

Table 5.3: Novel application mapping heuristic algorithm

Step 1: Construct a simplified overlay topology $\{\nu_{r'}\}, \{\lambda_{m'n'}\}$.	
Step 2: Repeat the following until $\{N_I\} = \emptyset$ or no solution can be found.	
Step 2.1	Order the unmapped application node list with a descending node degree and denote it as $\{N_I\}$.
Step 2.2	Choose the first node in $\{N_I\}$ and find a subset of candidate overlay nodes that (1) meet the degree requirement of the current application node; (2) can provide at least two substrate-diversified paths with no more than two intermediate overlay nodes to each application link connecting the current chosen application node to the already mapped application nodes; (3) have not been chosen to host application nodes before.
Step 2.3	Sort this candidate overlay node subset using the following criteria (in descending order) and notate the sorted list as $\{n_i\}$. (a) The average number of backup paths an overlay node can provide, capped at two; (b) Give preference to the overlay nodes that have already been chosen in the mapped solution; (c) Node quality (as measured by adding the inverse of the delay quality over all its associated QoS-compliant paths with no more than three hops in the simplified overlay topology).
Step 2.4	If $\{n_i\} \neq \emptyset$, choose the first overlay node in the list, remove the currently mapped application node from $\{N_I\}$; otherwise, add the previously mapped application node back to $\{N_I\}$ (i.e. backtracking) and go to Step 2. (Note: if no backtracking can be carried out, it will report no solution as stated in Step 2.)
Step 2.5	Compute the working paths for all the to-be-mapped application links connecting the currently mapped application nodes with the already mapped application nodes.
Step 2.6	Compute a backup path for each of the application links mapped in Step 2.5, ensuring no substrate-link overlap between the working and backup path pair.

5.5 Performance Evaluation

In this section, we first describe the evaluation metrics adopted here. Then, we present the simulation settings followed by the results for both the proposed ILP model and the new heuristic algorithm obtained through extensive simulations with synthetic topolo-

gies. Finally, we analyze the computational complexity of the proposed heuristic measured in terms of CPU time as compared to the best existing solution [16].

5.5.1 Evaluation Metrics

For evaluation purposes, we adopt metrics similar to those used in [16] and formally define each of them as follows:

1. D_{avg}: Average delay of the mapped application, namely,

$$D_{avg} = \frac{\sum_{\{L_{AB}\}} \sum_{\{\lambda_{m'n'}\}} D_{m'n'} W^{AB}_{m'n'}}{\mid L_{AB} \mid} \tag{5.15}$$

where $\mid L_{AB} \mid$ denotes the number of application links.

2. O_{wb}: It is defined as the percentage of application links that have a backup path sharing substrate links with the corresponding working path and is used to measure resilience of the mapped application request. In [16], it is defined as the ratio of the sum of direct and indirect paths over the number of direct paths. This does not consider the overlap of two mapped application links in the substrate layer. Moreover, we only consider the backup paths that are specifically allocated (i.e. path protection) since it usually takes a longer time if the backup paths have to be chosen from a pool upon failure of the working path(s) (i.e. path restoration) [86]. O_{wb} can be expressed as:

$$O_{wb} = \frac{\sum_{\{L_{AB}\}} O_{AB}}{\mid L_{AB} \mid} \tag{5.16}$$

where O_{AB} is binary and is equal to 1 when the number of substrate links the working and backup path allocated for an application link L_{AB} share is non-zero.

3. C_r: Resilience cost. It is defined as the number of overlay nodes that are used for backup but not as the hosting nodes for application mapping and is expressed as

follows:

$$C_r = \sum_{\{\nu_{m'}\}} B_{m'} \qquad (5.17)$$

5.5.2   Simulation Settings

The ILP models proposed here can be solved using different techniques. We only focus the performance of the enhanced ILP model solved using CPLEX [3] and denote it as *QRILP*. CPLEX [87] uses branch and bound techniques for solving ILPs and is capable of solving ILPs consisting of up to one million variables and constraints.

In order to evaluate the performance of the proposed model and the proposed heuristic (notated as *pQoSMap*), we also implement the latest heuristic *QoSMap* [16] [4]. In the *QRILP* model and the *pQoSMap* heuristic we have more stringent requirements than that of *QoSMap* in terms of finding backup paths. It might be impossible to find a solution. If there exists a solution, it is conjectured that the resilience cost will be higher than that of *QoSMap*. However, it depends heavily on the delay and connectivity features of the overlay candidate pools.

In order to evaluate the proposed ILP solution and heuristic method, we adopt network settings similar to those in [85] [5]. To be more specific, the substrate network has 80 nodes, generated using GT-ITM [88]. Each node pair is randomly connected with probability of 0.6, 0.8, and 1. These configurations are denoted as P60, P80, and P100, respectively, and the link delay is set by the topology generator. The overlay nodes are randomly attached to distinct substrate nodes in the substrate network with a delay of 1ms and the size of the candidate pool is fixed at 20. Depending on the application delay request, the simplified topologies have remaining connectivities from 40% to 100%. For an application request, we consider five different topologies [6], a full-mesh topology, ran-

---

[3]An explanation of the ILP model implementation is provided in Appendix B.

[4]The details of the simulation platform and its code verification are given in Appendix B.

[5]Other substrate networks, including another type of synthetic topology and a real-network dataset, are also considered later in Chapter 7 for a performance evaluation of the proposed heuristic.

[6]Please note that we use different application topology types to verify the performance of the proposed

domly connected topologies with 50% and 25% connectivities, a tree topology and a ring topology, each having 8 nodes. We carry out the simulations for all 105 scenarios with the *QoSMap* and *pQoSMap* heuristics. For the ILP model, we implement all the simulation scenarios except the ones where the application topology is a full mesh due to its high computational complexity. Unless otherwise specified, α is set to a comparatively high value whilst β and γ are set to a low value (expressed in (5.10)) to favour application mappings that can obtain better delay performance. $\alpha = 0.98, \beta = 0.01, \gamma = 0.01$ for the results presented in this chapter unless otherwise specified.

5.5.3   Results and Analysis

Firstly, we compare the performance of the three methods. Since the results are similar across different application topology requests, we only present the results for the simulation scenarios where the application topology is of 50% connectivity.

Figure 5.3 shows that the proposed ILP model leads to a much better averaged delay for the mapped application request and the average delay performance is comparatively stable under different application delay requests. Furthermore, the delay performance of the two heuristic algorithms are similar though there is some variation depending upon the requested application delay-bound. Note that when the substrate network is P60 and P80 and the remaining connectivity after pruning those links violating application delay constraint in the simplified overlay topology [7] is 40%, no solution exists for all methods. O_{wb} for the proposed ILP solution in all scenarios equals 0 while C_r equals 0 except in the $P80$ substrate network with the 70% remaining connectivity scenario, for which the value is 1 as depicted in Figure 5.4. By tuning the three weight factors in the ILP model to prefer the lowest C_r possible, no resilience cost for backup purposes can

---

ILP model and heuristics. This is similar to the scenarios adopted in [85]. The performance difference in accommodating various application topologies is not comparable since they have different features (i.e. different number of links).

[7] For brevity, the remaining connectivity is used to describe the simplified overlay topology in the rest of this chapter.
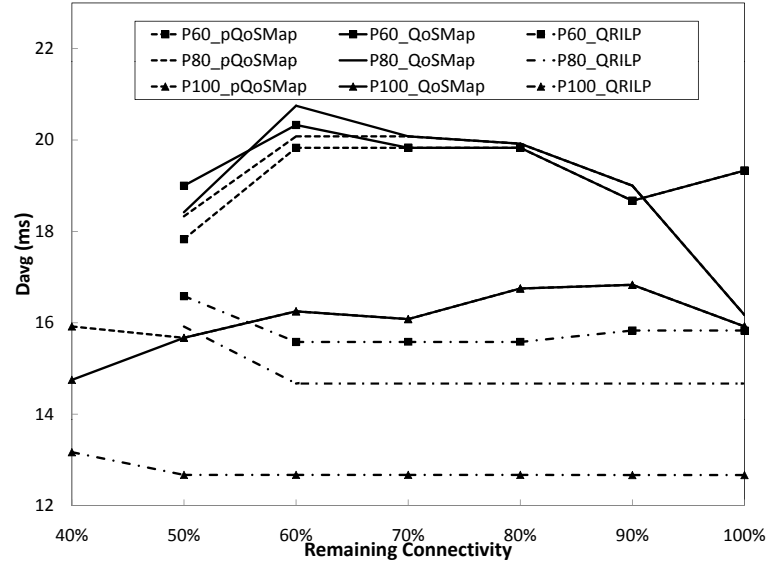
Figure 5.3: D_{avg} comparison for the $QRILP$ model, the heuristic $pQoSMap$ and the existing $QoSMap$

be obtained with a slightly increased value of resultant average delay [8]. On the other hand, the proposed heuristic can achieve none overlapped working and backup paths, namely, O_{wb} is equal to zero. However, this is obtained with additional overlay nodes included in the solution as shown in Figure 5.4. Although the $QoSMap$ heuristic seldom requires additional overlay nodes, it has much higher percentage of overlapped working and backup paths. This is illustrated in Figure 5.5.

Secondly, we compare the performance of the two heuristic methods in all the other simulation scenarios (i.e. application request with FM, 25% connectivity, ring and tree topologies). As shown in Figure 5.6, $pQoSMap$ needs to use additional overlay nodes to ensure diversified working and backup paths for supporting the application request. However, as the connectivity of the substrate topology increases, the number of additional overlay nodes decreases. Similarly, when the application delay requirement is less stringent, it costs less in terms of resilience cost to find non-overlapping backup paths. In general, an application request with a topology that is less-connected also reduces the necessity of using additional overlay nodes for the same purpose. Note that out of

---

[8]One example is presented in Appendix B.
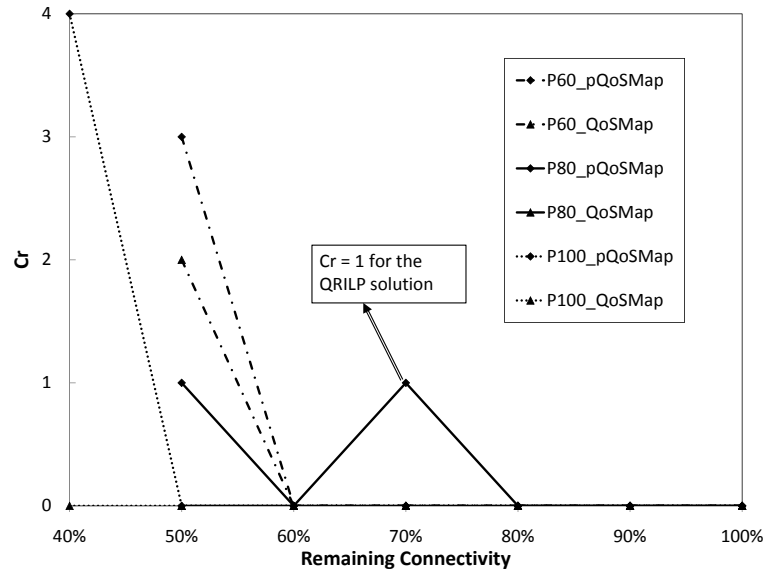
Figure 5.4: C_r comparison for the $QRILP$ model, the heuristic $pQoSMap$ and the existing $QoSMap$
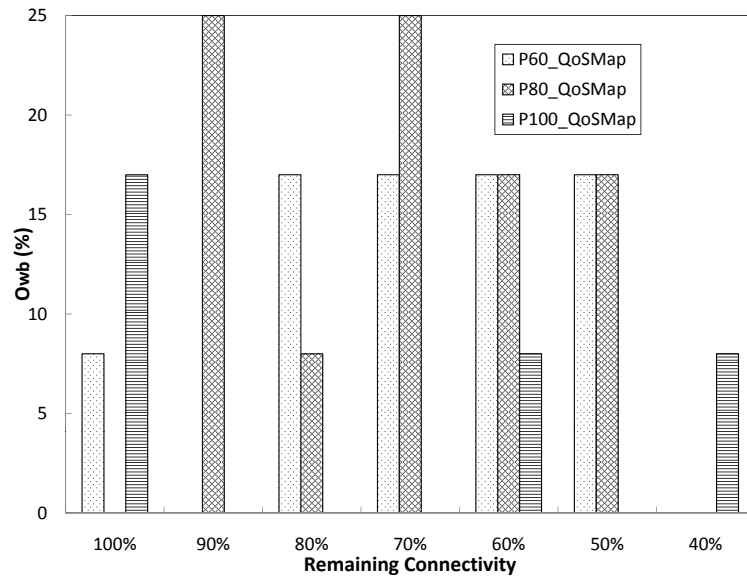
Figure 5.5: O_{wb} evaluation for the $QoSMap$ heuristic (Note: only non-zero O_{wb} values are shown.)

the 105 simulation scenarios we use, the $pQoSMap$ heuristic cannot obtain solutions in four cases although $QoSMap$ can. This is because no substrate-diversified backup paths exist in the overlay layer. Similar to the observation of the analysis given at the start of this section, the $QoSMap$ method performs worse in terms of resilience as compared to
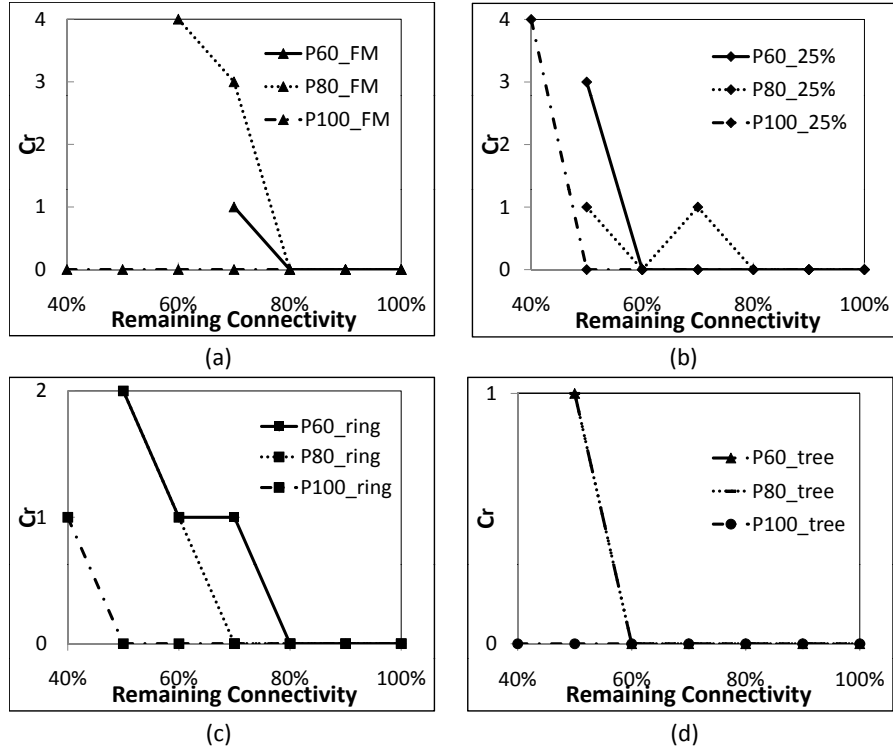
Figure 5.6: C_r evaluation of the $pQoSMap$ heuristic with FM, 25%, ring and tree application topologies and different delay requirements

the proposed heuristic since it incurs a much higher percentage of overlapping working and backup paths whilst $O_{wb} = 0$ for all $pQoSMap$ solutions. This is confirmed by the simulation results presented in Figure 5.7.

Furthermore, we compare the average delay achieved by both heuristic methods using the D_{avg} Ratio. This metric is defined as the D_{avg} of the $pQoSMap$ heuristic divided by that of the $QoSMap$ solution. The results are illustrated in Figure 5.8. As shown in the graph, the average delay achieved by both methods are comparable in all simulation scenarios [9]. In general, the $pQoSMap$ solution needs to sacrifice delay performance when the delay requirement of the application is too tight. Note that in some scenarios $pQoSMap$ performs better than $QoSMap$ in terms of the average delay of the mapped application request. This can arise since $QoSMap$ cannot guarantee to choose nodes with

---

[9]The difference between these two heuristics is smaller than 25% in almost all cases. Moreover, both methods can achieve much lower average delay than required by the application mapping requests.
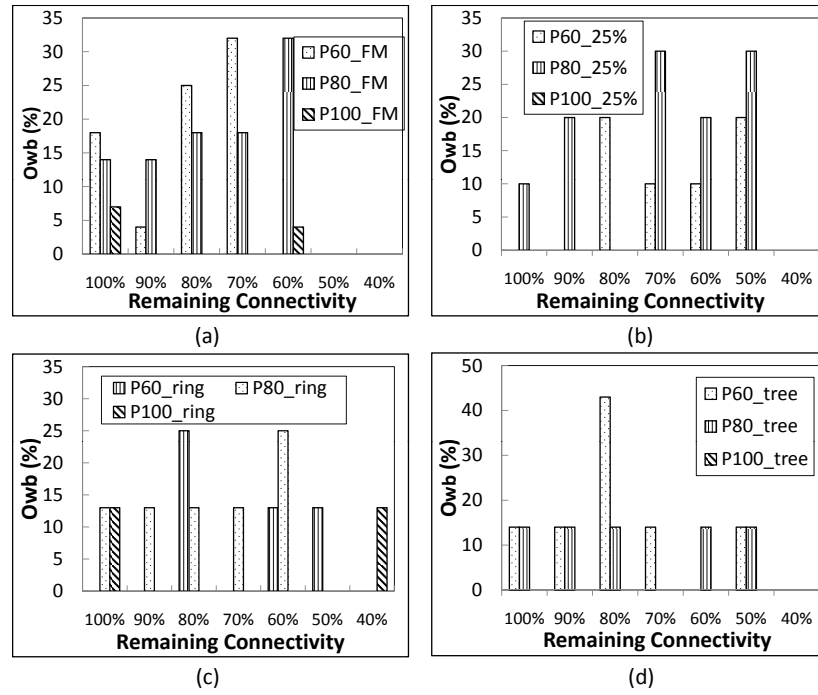
Figure 5.7: O_{wb} evaluation of the *QoSMap* heuristic with FM, 25%, ring and tree application topologies and different delay requirements
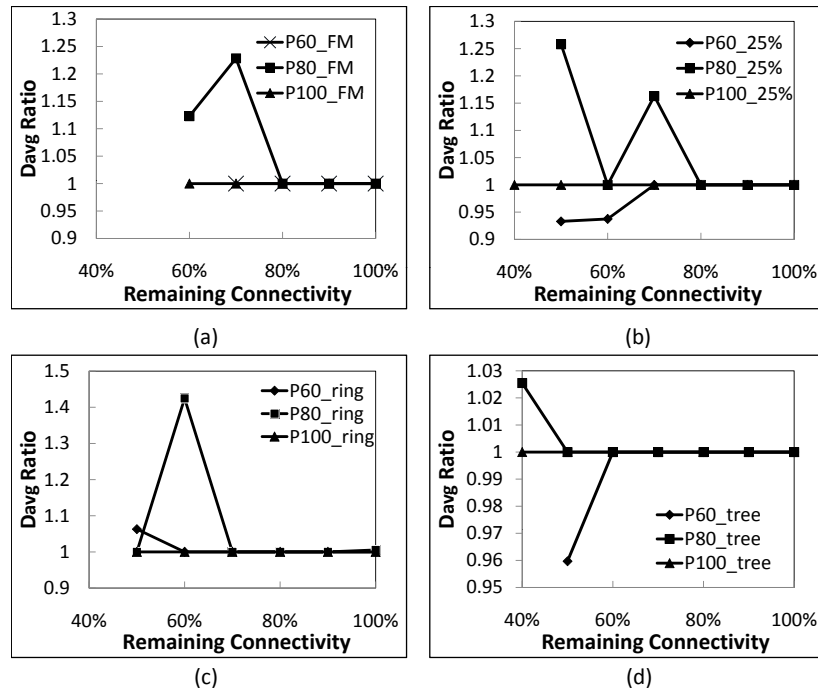
Figure 5.8: D_{avg} Ratio evaluation of the *QoSMap* and *pQoSMap* heuristic algorithms with FM, 25%, ring and tree application topologies and different delay requirements

the best delay performance since it chooses overlay nodes sequentially with a preference for previously chosen overlay nodes. Since the *pQoSMap* heuristic chooses backup paths exploiting substrate topology information, its solution can select a set of overlay nodes with a lower average delay.

5.5.4 Computational Time Analysis of the Novel Heuristic

In order to compare the computational complexity of both algorithms [10] (based on an Intel-based machine with a 2.4 GHz processor), we adopt the P60 topology employed in the previous section and attach one overlay node to each of the substrate nodes with 1ms delay (i.e. with 80 overlay candidate nodes). The node size of the application request varies from 10 to 50 with an interval of 5 and the remaining connectivity of the simplified overlay topology varies from 50% to 100% depending on the settings of the application delay request. The application topology type is the same as described in the last section.

For all the scenarios, when a mapping solution exists, the *QoSMap* heuristic takes less than 1 second to find one solution. Similarly, the *pQoSMap* heuristic takes less than 1 second except for 3 scenarios where the CPU time is 2 seconds. In the three scenarios, the application topology request is a full mesh with 40, 45 and 50 nodes, and the remaining connectivity of the simplified overlay topology is 100%. However, if no solution exists, both algorithms have a high execution time since considerable backtracking arises to iterate all eligible candidate nodes before reporting no solution. The running time when no solution can be found for an application topology with 25% connectivity is shown in Table 5.4. One possible solution to avoid long execution times is to assign a limit and stop searching once this threshold is reached. In summary, the execution times of the proposed heuristic are comparable to those of the *QoSMap* heuristic.

---

[10]We do not compare the computational time of the ILP solution since it is infeasible for the settings we adopt here.

Table 5.4: Execution time (in seconds) comparison when no solution exists for the application topology with 25% connectivity

Overlay Size	Remaining Connectivity	$pQoSMap$	$QoSMap$
50	60%	414	285
	50%	377	284
45	50%	381	300
40	50%	389	297
35	50%	12	258

5.6 Summary

In this chapter, a novel overlay mapping model exploiting Integer Linear Program is proposed that can provide enhanced QoS performance and effective resilience. In order to improve the effectiveness of the backup path selection, substrate network topology information is taken into consideration. Through small-network simulations, it is verified that the proposed ILP model can perform much better than the heuristics considered.

Although solving the ILP for larger networks is infeasible, it still provides a useful benchmark for evaluating existing and new heuristic algorithms with small networks. For solving the problem with large networks, a novel heuristic is proposed that exploits the substrate information to obtain a feasible solution. Through simulations with synthetic networks, it is verified that the proposed heuristic can provide more effective backup paths compared to the state-of-the-art best solution. However, it does not necessarily provide the best quality of service and a small number of additional overlay nodes may be needed for effective backup purposes.

Chapter 6

Survey of Substrate Topology Inference through Active Probing

6.1 Overview

In the last two chapters, we have shown that substrate topology information is instrumental to secure better performance from the ROMCA overlay for providing both resilience service and application mapping. Although there are surveys summarizing the efforts toward discovering the Internet topology, no work has addressed how to infer the routing topology among a particular group of hosts [1] scattered across the Internet. These two issues differ in scale, objective and solutions currently available. How to infer the routing topology among a group of hosts is non-trivial due to the complexity, size and decentralized nature of the Internet. Nevertheless, a cost-effective means of inferring topology would be of considerable benefit to the ROMCA overlay as well as other applications summarized in Chapter 2. Since external information, such as routing table entries, is not publicly available to hosts, most of the methodologies employ an active probing

---

[1] "Hosts" are defined as the set of machines that are under control of the routing topology discovery mechanism or cooperate in order to provide this function. Hosts can be either end-systems or routers. In this thesis, it refers to the overlay nodes in ROMCA.

mechanism to address this issue. For this reason our focus is on active probing based solutions, too.

In this chapter, a comparison between generic Internet topology discovery and that operating among a specific group of hosts is examined in order to distinguish these two situations. Furthermore, we discuss the motivation that underlines the significance of various techniques addressing Routing Topology Discovery (RTD) among a group of hosts. We then classify the strategies into two types: Router-Assisted (RA) and Non-Router-Assisted (NRA), and consider in detail the state-of-the-art position, including the merits and challenges of various approaches. A discussion and summary are then presented in the last section.

6.2 Problem Description

We summarize the main strategies proposed in the literature for inferring the network-layer topology [2] among a group of hosts scattered across the Internet. It is of significant relevance to applications which involve a two-layer network where the lower layer is the Internet and needs *a priori* knowledge of the (inferred) lower-layer topology to obtain better performance for the upper layer. For example, this information is essential to secure better performance for the applications supported by the ROMCA overlay in this thesis. In the rest of the chapter, the term "node" and "router" will be used interchangeably to describe the nodes in the inferred topology other than the hosts. Moreover, topology discovery mechanism among a group of hosts will be referred as "RTD-Selective" whereas the topology discovery across the whole Internet will be termed as "RTD-Complete" for brevity.

A network-layer topology can describe a network at various levels, such as the inter-face level, router level [3] and AS level [33]. If combined with information from other layers,

---

[2]The terms network layer topology, routing topology, and layer-3 topology will be used interchangeably throughout the chapter.

[3]It is also referred as IP layer in other literature, such as [32].

other types of topologies can also be obtained. For example, combined with geographical information of the routers, a Point-Of-Presence (PoP) map can be obtained [32]. Due to the limitation of active probing, this chapter predominantly focuses on the interface level and router level topologies [4] and presents a systematic overview of routing topology discovery mechanisms through active probing that operate among a group of hosts.

There are two main strategies for solving the RTD-Selective problem through active probe injection: traceroute-based and tomography-based. Both exploit active probes to obtain information to infer the topology representing network-layer connection relationships among a group of nodes, often residing across multiple domains. The traceroute-based [5] methods send messages, e.g. ICMP/UDP packets [70], among the group of hosts to infer the topology. This type of method is simple but depends on the routers response to the probe messages where appropriate (e.g. TIME EXCEEDED reply with detailed information) and additional procedures are usually needed because non-standard router behaviours are common in the Internet as discussed in Section 6.3. This strategy is also termed Router-Assisted (RA).

On the other hand, the tomography-based approaches operate without the necessity of router responses, but rely solely on collected end-to-end information. These are termed Non-Router-Assisted (NRA). However, they may need to retrieve a substantial volume of data in order to construct an accurate representation of the topology. Moreover, the computational complexity of tomography-based strategies is comparatively high compared to traceroute-based ones since statistical and signal-processing techniques are usually employed within the inference process [91]. Finally, NRA methods can infer only the logical topology, which is a simplified representation of the actual substrate topology. To be more specific, a single logical link in the inferred topology represents the set of

---

[4]Topologies at other levels require relevant information from network providers either directly or indirectly. For example, inter-domain routing tables provided by Internet Service Providers are needed to obtain the AS-level topology, or publically available databases [89] can be exploited. We refer interested readers to publications [33, 34] for further information.

[5]In this chapter, the term "traceroute" refers to all the probe methods that take advantage of Internet Control Message Protocol (ICMP) based discovery with or without modifications. The term "Traceroute" is used to specifically present the standard "Traceroute" tool specified in ICMP protocol [90].
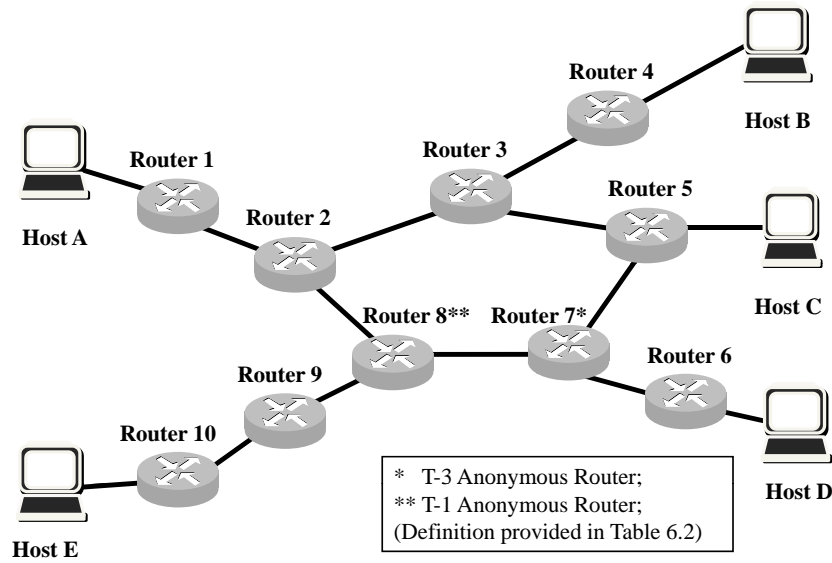
Figure 6.1: A network topology example
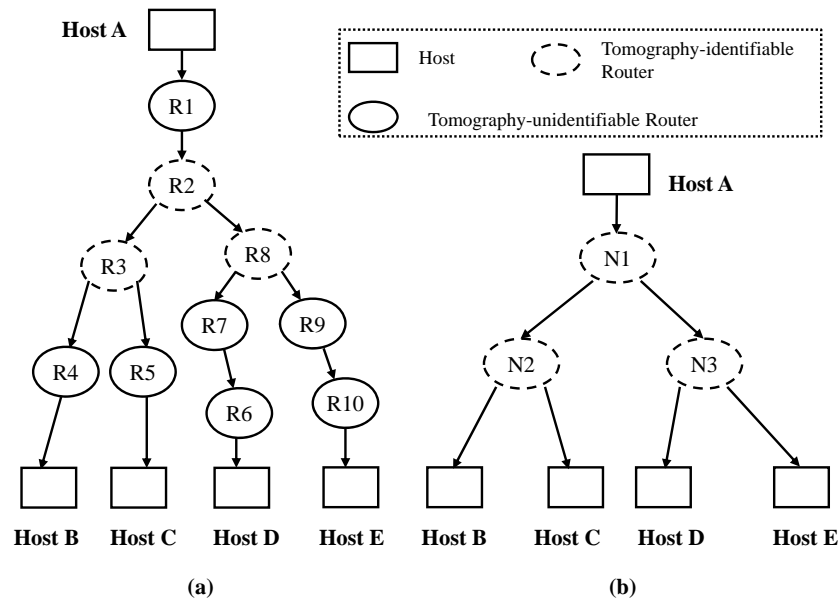
Figure 6.2: Topology discovery example: (a) the topology obtained using the traceroute-based method in an ideal scenario where all the routers behave in the standard way; (b) the logical topology obtained using the tomography-based method

substrate links connecting two branching routers in the substrate topology. For example, both methods are deployed on the network topology depicted in Figure 6.1, where Host A is the source and all the rest of the hosts are destinations. The inferred topology obtained using the RA methodology, assuming all routers responding to traceroute mes-

sages, will be the topology as shown in Figure 6.2(a). However, the tomography-based strategy can only obtain a simplified logical topology as depicted in Figure 6.2(b), as the non-branching internal nodes are not identifiable.

Although RTD-Complete and RTD-Selective share many similarities, as stated in this section and Section 6.3, there are several aspects that differentiate RTD-Selective mechanisms from existing efforts in obtaining the Internet topology. Firstly, the impetus behind work addressing these two issues is not the same. Internet topology inference is of crucial significance to a systematic understanding of the current Internet, its evolution and modeling [32] as well as facilitating resilience and tackling security issues [33]. On the other hand, network topology inference among a group of hosts is application-oriented and often conducted with comparatively few resources. The inferred topologies in the RTD-Selective problem are of interest to activities that usually need this information obtained in a timely fashion. This information can be used to either improve the performance of proposed algorithms or enable engineering/strategies based on the knowledge about the part of network that is concerned. Secondly, the scale of the targeted topology for these two problems is drastically different. The RTD-Complete solution aims to find the routing topology of the whole Internet and usually takes a long time, e.g. of the order of days [66], to execute the discovery process. The usual active probing methods use probes that need router responses, such as traceroute [32]. Whereas, the number of nodes involved in the RTD-Selective case is much smaller and it is possible to exploit techniques such as network topology tomography to solve this problem. Thirdly, RTD-Complete schemes usually exploit a small number of Vantage Points (VP) to a large pool of publicly routable IP addresses that they have no control over. That is to say, strategies address this RTD-Complete issue by sampling the Internet using traceroute tools. The inferred topology obtained in such a manner is proved not to be able to acquire a complete router and link set for the Internet (i.e. this is often referred to as the sampling bias problem), thus it cannot represent the true features of the Internet. For example, the power-law characteristic of the Internet is questioned [92, 93]. Another

issue with using tracerouting for RTD-Complete originates from the asymmetrical routes found in Internet; Reverse traceroute [94] is proposed to solve this. Quite differently, these are not issues for RTD-Selective solutions because (1) they only endeavor to discover the topological relationship of the routers covered by the paths connecting each pair of hosts, not the complete Internet topology; (2) forwarding and reverse traceroute paths between a pair of hosts are obtainable since all the entities that can carry out active probing are either under the control of the RTD-Selective discovery mechanism or can cooperate with each other. A discussion about routing topology inference of the Internet at various levels is beyond the scope of this work and interested readers are referred to the survey works [32–34].

6.3   Router-Assisted RTD-Selective

RA topology discovery among a group of hosts includes three processes: E2E probe injection, initial data processing and AR router resolution [6]. The second process includes integrating the traceroute entries, dealing with anomalies (such as presence of private addresses) and resolving router aliases [95] (i.e. mapping IP interfaces to routers). According to how the first two processes interact, RA RTD-Selective solutions can be categorized into sequential or iterative. The basic flow diagram of these alternatives is shown in Figure 6.3. In the sequential methods, as indicated by the name, the probe injection and initial data handling are performed serially, while these two steps are done repeatedly in iterative methods.

Using either of these methods, only an initial interface-level topology is obtained. It is generally a redundant topology in non-ideal cases as a result of traceroute measurement artifacts. A typical example is that a router which does not respond to traceroute messages i.e. an Anonymous Router (AR), may be represented by multiple nodes notated with different identifications. As shown by the examples in Chapter 2, the router-level

---

[6]This is often the case when traceroute is employed for use in the Internet. However, this step is not included in ideal cases.
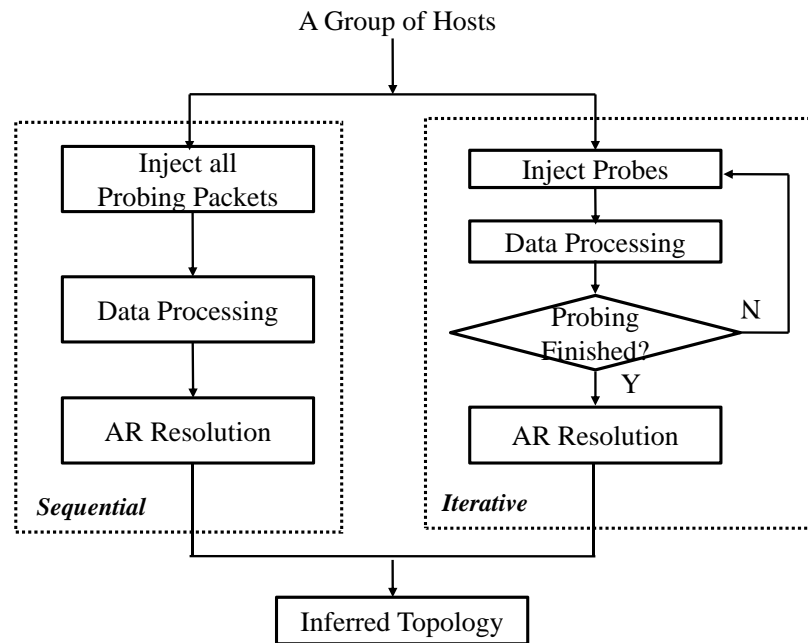
Figure 6.3: Workflows for the RA RTD-Selective process

topology is usually required. Thus, merging techniques are also needed to fuse the IP address of each router interface into a single router identifier so as to obtain an accurate router-level topology as possible. Moreover, probing overhead of pair-wise tracerouting increases quadratically with the number of involved hosts. Hence, the main issues of this inference method that have been researched are: (1) traceroute measurement artifacts and how to prevent/mitigate them; (2) the techniques to reduce the probing cost during the injection process; (3) the algorithms used to solve the anonymous router problem and (4) IP Alias resolution. Since some of the issues are common to that of RTD-Complete schemes and previous Internet topology surveys [32–34] have already covered some aspects of this, we will only elaborate those elements that are either new or closely related to the problem addressed here but not for the RTD-Complete case in this section. However, a brief description of some important methodologies designed for the RTD-Complete case is included here for completeness.

6.3.1 "traceroute" Basics

One of the main features associated with the RA approach is that it relies on routers responding to the probes issued by the peripheral hosts, in addition to their basic forwarding function. The most well-known tool is Traceroute written by Van Jacobson for network debugging purposes [96].

Assume that Host A Traceroutes to Host B in the network topology shown in Figure 6.1. In this instance, Host A issues UDP messages (or ICMP echo Requests) with increasing value of Time-To-Live (TTL) starting from 1. When intermediate routers, for example Router 1, receive the packet, they decrement the TTL field by 1 before forwarding it to the next-hop router. If in doing so, the TTL value reaches zero, then the router will discard the message and send back a "TIME EXCEEDED" error message to the source node, including its own IP address. Thus, Host A will learn the IP address of the intermediate routers from the reply messages obtained in turn, together with the Round Trip Time (RTT) that it can measure. Finally, the UDP message is delivered to the destination node (i.e. Host B in this example). The destination node will reply with a "Destination Unreachable" message with error code "Port Unreachable" as the received message is usually assigned with an unused high port number. Thus, the complete route traversed by packets sent from Host A to Host B is obtained. This process is illustrated in Figure 6.4 [33].

With the growth, commercialization and distributed autonomy of the global Internet, these Traceroute-based tools face feasibility and accuracy challenges. Table 6.1 itemizes the various types of traceroute tools currently available. Although some of them will be discussed later in this section, we refer the reader to the schemes listed in Table 6.1 for more information.

For example, an increasing number of routers are configured by operators [103] to behave in a "non-standard" way as specified in [90] and appear as "*"s in the Traceroute output. This type of router is generally referred to as an anonymous router. For
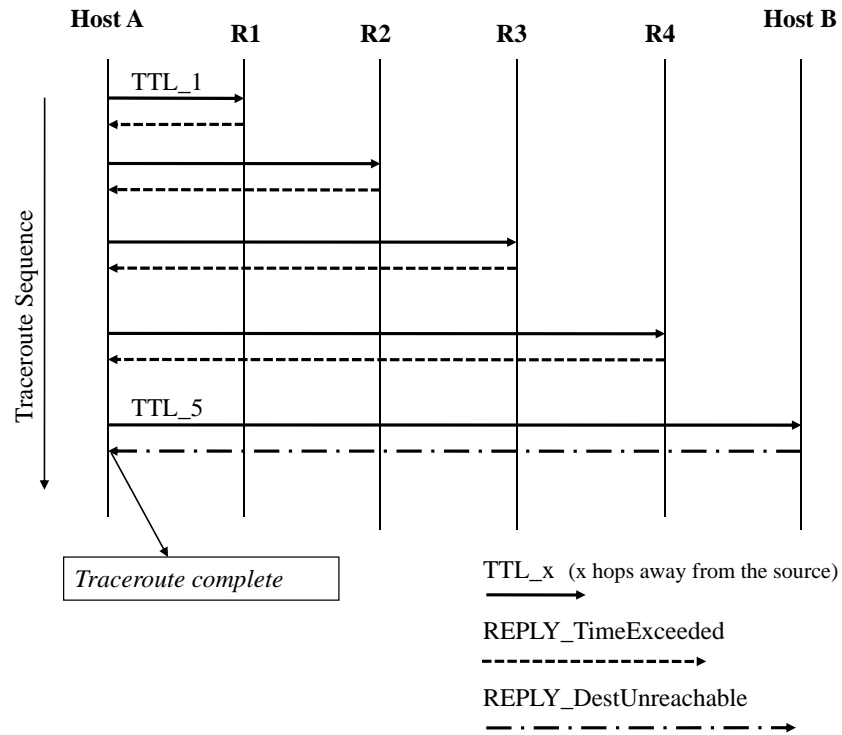
Figure 6.4: Traceroute process

Table 6.1: Summary of "traceroute" tools

Probe Type	Project Example	Cost	Auxiliary Methods
ICMP based	Skitter [36]	Low	None
UDP based	DIMES [97]	Low	None
TCP based	—	Low	Using TCP SYN packets [33, 98]
Doubletree based [99]	Max-Delta [100]	Medium	End hosts need to share traceroute results.
Paris traceroute [101]	Archipelago [66]	Medium	Certain fields in the packet header are manipulated, such as the "sequence number". [102]
Reverse traceroute [94]	—	High	Source spoofing, IP timestamp, record router options and use of multiple vantage points

instance, some routers may not respond to and even discard the probe messages. So, the

source cannot retrieve any information about the Round Trip Time (RTT) and addresses

subsequent to this un-responsive node along the forwarding path. In order to avoid the

excessive sending of probes to such un-responding routers, the maximum hop length is set to 30 by default. Another example is that the behaviour of routers may depend on the probing rate, which is known as a rate-limiting configuration. To be more specific, a router will not reply if the rate is higher than a pre-defined threshold, otherwise it will respond. The classification and definition of ARs are summarized in Table 6.2, together with examples. These non-standard router behaviours are quite common in the Internet as discussed in [99, 103, 104] using real traceroute datasets.

The current *de facto* operation of the Internet also has negative effect on the credibility of Traceroute results. Firstly, some ISPs may adopt load balancing in their networks so as to improve the overall performance. This results in the phenomenon that data packets of a single application/flow may not take the same route. This affects Traceroute probes too and leads to inaccurate router/link path information. Secondly, the route taken from the source node to the destination node may not be the same as the reverse route [3, 94]. Therefore, traceroute can only obtain the source-to-destination path information, leaving the destination-to-source one unexplored. Moreover, in order to improve the network performance, many ISPs now deploy Multi-Protocol Label Switching (MPLS) on their routers. Instead of looking at the IP header of the incoming packets, routers forward the packets much faster by inspecting a shorter fixed length label identifier associated with the packets. Therefore, the TTL field may not be reduced on a hop-by-hop basis. Inaccurate information of the path may thus result. To avoid this anomaly, the only existing solution is to exploit enhanced traceroute [105], which is informed by the existence of label-switch routers to factor in the appropriate hop count.

To tackle the first issue, B. Augustin *et al* propose an improved traceroute tool called Paris traceroute [101]. It can force all sequential probes to follow a single path by manipulating the fields in the probe header. This is based on their experimental observation that a natural flow identifier is the classic five-tuple of fields from the IP header and either the TCP or UDP headers: Source IP address, Destination IP address, Protocol, Source port, and Destination port [102]. For example, the sequence number and ICMP

Table 6.2: Anonymous routers classification and definition

Type	Definition	Example (using Figure 6.1)
T-1	This type of AR does not reply to ICMP messages, but it will forward the messages to next-hop node.	**From Host A to Host E:** 1: normal [R1 IP Address] 2: normal [R2 IP Address] 3: ★ ★ ★ Request Timeout ⇒ [R8] T-1 4: normal [R9 IP Address] 5: normal [R10 IP Address] 6: normal [Host E IP Address]
T-2	This type of AR replies to ICMP messages depending on its working status, namely it may reply in light load but not in heavy load.	**Case 1:** . . . . . . X: 30ms 40ms 35ms [IP address] . . . . . . **Case 2:** . . . . . . X: ★ ★ ★ Request Timeout ⇒ T-2 . . . . . .
T-3	This type of AR will discard ICMP messages without replying and forwarding, thus results in all the routers downstream staying unknown to the source.	**From Host C to Host D:** 1: normal [R5 IP Address] 2: ★ ★ ★ Request Timeout ⇒ [R7] T-3 3: ★ ★ ★ Request Timeout 4: ★ ★ ★ Request Timeout . . . . . . 30: ★ ★ ★ Request Timeout
T-4	This type of AR will respond to ICMP messages if traceroute rate is below certain limits, otherwise it will discard the messages.	Similar to that of T-2, but the routers will have to monitor the traceroute rate and respond to the ICMP messages. If the rate detected is higher than the pre-specified value, it will behave as shown the same as Case 2 for T-2. Otherwise, it will behave as Case 1 for T-2.
T-5	Private IP addresses present in the traceroute path. They should not appear as private IP addresses do not provide meaningful information for public use.	X: 30ms 40ms 35ms 10.1.2.5 ⇒ T-5 X+1: normal [X+1 IP Address] X+2: normal [X+2 IP Address] X+3: normal [X+2 IP Address]

header for ICMP-based probes are manipulated to keep the flow identifier unchanged. Thus, all the probe packets to the same destination will be sent over the same paths.

This can avoid some of the anomalies encountered during the probing process and obtain accurate path information [106]. Furthermore, they expand this traceroute mechanism to discover all the possible multiple paths between a source-destination pair [107, 108].

On the other hand, in order to solve the asymmetry issue, "reverse traceroute" is proposed by E. Katz-Bassett *et al* [94]. The auxiliary strategies required to perform the reverse path discovery process include exploiting the IP timestamp and Record Route (RR) options in IP header extensions, together with source spoofing by exploiting multiple vantage points. The basic process is briefly described as follows. The vantage points will send packets spoofing the source node to the destination with RR and/or Timestamp options employed. Thus the source node can obtain complete/partial [7] information about the route from the vantage point, via the destination node and back to itself. Given the paths from vantage points to the source node are known, the source can complete the reverse path discovery from the recorded route information obtained during the spoofing process. As this approach involves a complex procedure to infer the reverse path from the destination to the source, it is still debatable whether it will be widely adopted [109]. Moreover, this asymmetry problem of Internet paths is not an issue for RTD-Selective schemes since all the participating hosts are assumed to be controlled by a common organization or operate cooperatively. In this case, the reverse traceroute strategy can be readily used to determine the omitted part of the route information if the unresponsive router is present within the first several hops of the usual tracerouted path.

6.3.2 Probing Cost Reduction

Given N hosts and assuming symmetric paths between each pair for simplicity, if pairwise traceroute is conducted, $N(N-1)/2$ traceroutes are needed. How to reduce this probing overhead has been researched in order to achieve efficient topology discovery. Since this is also a problem in RTD-Complete schemes, techniques to reduce the probing

---

[7]The IP optional header supports the recording of a limited number of IP addresses. So if the route to be recorded is too long, then only partial information can be noted.

overhead for them can also be utilized here. However, their effectiveness needs to be re-evaluated since the issues discussed here aims to traceroute between N hosts instead of carrying out an $N-to-M$ tracerouting, where N is the number of monitors and M is the number of the destination addresses and these two values are highly unbalanced [66].

In reducing the traceroute overhead in discovering the interface-level Internet topology, B. Donnet *et al* take the initiative verifying the existence of high redundancies between traceroute entries through quantitative analysis [99]. Their discussion relies on the assumption of tree-like routing structures in the Internet. The Doubletree algorithm [99] is proposed to reduce two types of probing redundancy, namely: intra-monitor and inter-monitor redundancy. The former means there are duplicated visits to routers if traceroute is initiated from a single source to multiple destinations. And the latter denotes the multiple visits to the same router when multiple sources discover routes to the same destination. It is by manipulating the traceroute sequence as well as sharing the information among these monitors that they either reduce both of these redundancies or one of them. Detailed analysis is covered in [32] and [33], and thus omitted here.

In another independent work [100], Xing *et al* also carried out experiments on measuring pair-wise traceroute redundancy. Unlike B. Donnet's work, in the network scenarios considered by Xing, there is no clear boundary between source and destination among a group of hosts. Moreover, they assume the measurements are based at the router level instead of the interface level. Nevertheless similar results are obtained showing high redundancies within the Traceroute results.

Based on their analysis, Xing *et al* propose a landmark-based Traceroute method called Max-Delta to discover the topology among a group of nodes with a reduced number of Traceroute probes [110, 111]. The main idea is that if there is a large difference between the Euclidean distance obtained through N landmarks and that of the inferred topology for a pair of hosts, then it is highly probable that the path between the two hosts contains many unexplored routers. Therefore, they prioritize the sequence of Traceroutes for each host, in order to discover most of the incident links and routers in fewer iterations with

reduced overhead compared with traditional approaches. In their later work, they expand the scheme enabling it to be implemented in a distributed way. They also incorporate the Doubletree algorithm and employ a router ID mapping table to further reduce overhead and complexity [100]. More recently, they extend the Max-Delta scheme by incorporating a preference for paths with shorter delays to reduce the traceroute resource consumption and complexity whilst maintaining the inference accuracy [112]. In a separate work, they propose a scheme that does not rely on coordinates [113]. However, in all their studies addressing the probe reduction issue, they assume no ARs are present in the network. Moreover, they do not mention how they can obtain the router-level topology. Although this can be solved by the techniques summarized in Section 6.3.4, the effectiveness of their proposed methods needs to be examined since additional probing traffic may arise during the IP alias resolution process.

6.3.3 Anonymous Router Resolution

The presence of anonymous routers in a network greatly inflates the topology collected by pair-wise traceroutes. To be more specific, T-1 ARs result in a "*" entry for each traceroute that goes through them whilst T-3 ARs makes all the routers remain undiscovered and thus all are represented by "*" entries. Each of the "*" entries is normally treated as an independent router and thus generates a greater number of routers and links in the inferred topology than truly exist. A simple example is depicted in Figure 6.5. The number of nodes and links in the aggregated topology without AR resolution, as shown in Figure 6.5(b), is greatly inflated as compared to the original one as illustrated in Figure 6.5 (a). Thus, it will result in an inaccurate representation of the true topology and AR resolution techniques are needed to decrease the number of duplicated nodes and links. It has been proved that this AR resolution issue is NP-hard [114] and heuristics are usually proposed to solve it in a time-efficient manner. Recently, research in [115] has formally proved that there is no algorithm that can guarantee to obtain a set of candidate topologies that includes the original topology in the presence of even a small
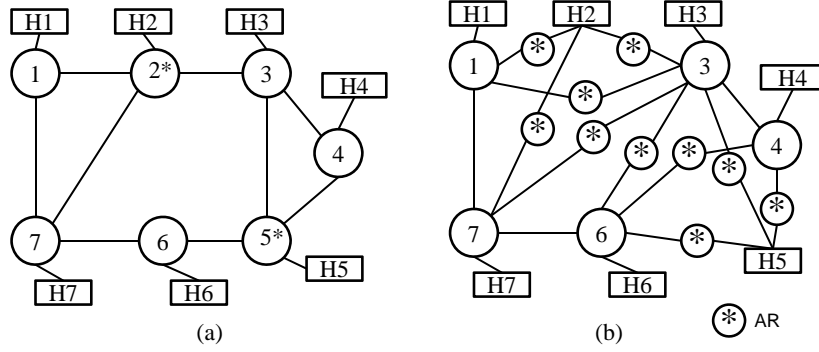
Figure 6.5: An example of AR presence impact in topology discovery: (a) actual network; (b) inferred topology without AR resolution

percentage of anonymous routers.

Only a moderate amount of work addresses this AR presence problem, among which Yao *et al* [114] pioneered the study of merging anonymous routers to create a topology that is as close to the real topology as possible. They propose a heuristic solution. However, their heuristic for checking whether two routers are mergeable has computational complexity of $O(n_a^4)$ [110] [8], where n_a is the number of anonymous routers presented in the aggregated topology. Observing that this method would only be applicable in networks of small/medium size, Xing *et al* [110] proposed two methods with less complexity to make the solution more scalable. The first one utilizes a generalized multi-dimensional scaling technique to merge nodes with similar multi-dimensional coordinates. This method has computational complexity of the order of $O(n_a^3)$. In order to further reduce the complexity, they propose a much simpler heuristic called the neighbor matching (NM) algorithm, which trades off accuracy for lower complexity of the order of $O(n_a^2)$.

The most recent strategy proposed by M. Gunes *et al* [104] exploits Graph-Based Induction (GBI) to resolve each type of redundant structure separately. More specifically, they visualize the graphical structures exhibited in the inflated topology and reduce the

---

[8]In the redundant topology of large-scale networks, the number of ARs is the main factor that determines the computational complexity and is considered in this chapter. Other factors, such as the number of known routers and end hosts, can also be considered as discussed in [110].
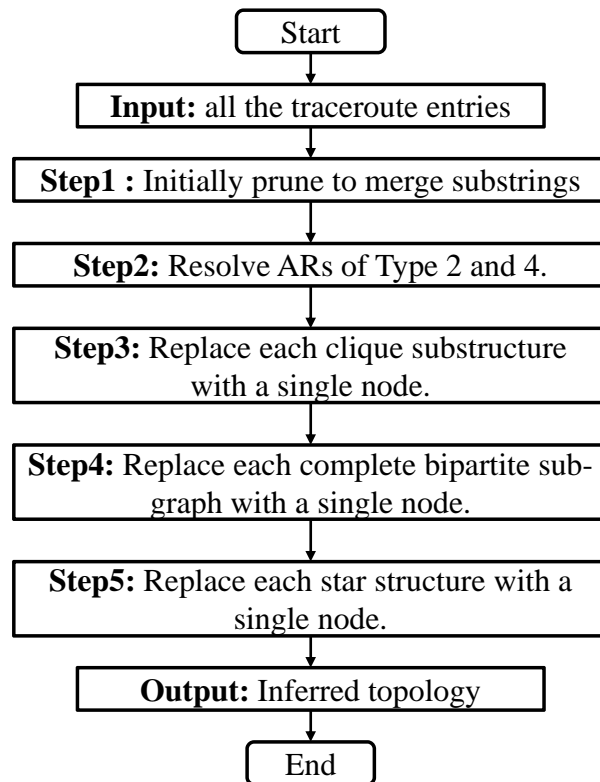
Figure 6.6: Workflow of the GBI AR resolution algorithm

number of anonymous nodes by replacing the structure with the original anonymous routers [9]. They execute five algorithms/steps in sequence as depicted in Figure 6.6 and the computational complexity is determined by the five steps and is proved to be much lower than that of previous proposed algorithms. To be more specific, according to their analysis on a dataset obtained from the iPlane project [39], the actual computational complexity of their proposed scheme for this instance is three orders of magnitude less than that of the previously best approach (i.e. the NM algorithm). A comparison of all the topology inference methods is provided in Table 6.3.

---

[9]One single anonymous router may be represented by multiple anonymous nodes as shown in Figure 6.5.

Table 6.3: Algorithms for AR resolution in topology inference

Methods	Compu. Complexity	Accuracy[10]	AR types	Description
Simple Merging	$O(n_a^4)$	Low	T-1	Merging obeying distance preservation and trace consistency principles[11].
ISOMAP Based	$O(n_a^3)$	High	T-1, T-3, T-5	Capturing the correlation of high dimensional data in low dimensional space and merging closely located nodes.
Neighbour Matching	$O(n_a^2)$	Medium	T-1, T-3, T-5	If two ARs share at least one neighbour and do not appear in same trace, they will be merged.
GBI Algorithm	$<O(n_a^2)^{note12}$	High	All Types	Merging ARs by finding certain structures in the inflated topology. [104]

6.3.4 IP Alias Resolution

As explained previously, routers typically have multiple interfaces, each assigned with different IP addresses. Therefore, even though multiple traceroute probes traverse a single router, it might be represented in those traces by its different interface IP addresses. The strategies for solving this problem in order to obtain a router-level topology are termed IP alias resolution [116]. As analyzed in this work, it is shown that the graphical properties of inferred topology can be highly distorted if IP alias resolution has a low success rate. It also indicates that, with a low IP alias resolution rate, the topology inferred in the RTD-Selective case suffers more than unbalanced sampling, i.e. the methods deployed for discovering the Internet topology.

---

[10] Accuracy is evaluated using "Edit Distance" here, which represents the similarity between the original and inferred topology. Edit Distance is defined as the number of steps (node addition/deletion, link addition/deletion) to transform the inferred topology to the real topology. Other metrics for evaluation can also be found in related papers.

[11] Distance preservation means the AR resolution process should not reduce the length of a shortest path between two nodes in the resulting topology whilst trace consistency means two ARs included in a single trace cannot be merged for the sake of inference accuracy.

[12] The exact value depends on the size of the input topology for each step of the proposed algorithms as analyzed in [104].

As summarized in Table 2 of survey [33], there are generally two principle approaches that can be exploited. The first type is based on sending additional probe traffic to test if two potential IP addresses belong to a single router or not. Moreover, most of the schemes in this category rely on routers responding to the probe messages. This is not feasible if the router probed is an AR. Recently, Santi *et al* proposed better types of probe packets which can improve the performance of existing methods, i.e. Mercator Identification and Ally [117]. Another recent work exploits the IP timestamp option [118]. The basic principle is to send a packet requesting a timestamp value for two suspected IP aliases. They found that their proposed method can achieve reasonable performance and is able to find alias pairs that were not found with previous methods.

The second principle approach is based on analytical strategies using tracerouting results using the convention of IP addresses assignment in the Internet [116]. This method identifies IP address pairs of a point-to-point link in a forwarding and reverse path pair of two hosts so as to infer possible IP aliases in those traces. It has two advantages. One is that no additional traffic is injected and second is that it can be done offline after all the tracerouting is finished. Although this method might not be easily deployable for Internet topology discovery since the destination nodes are not normally under control, it is not an issue for the scenarios we discuss here since all the hosts are assumed to be either under control by a single entity or at least operate cooperatively. However, the effectiveness of this method suffers because of the incomplete route information obtained due to unresponsive routers. As suggested by [116], a combination of both strategies can be employed in order to improve the IP alias resolution success rate.

6.3.5 Limitations and Issues

Although traceroute-based probing is simple and has been widely implemented for topology discovery, there are also other efforts assessing the validity of the assumed characteristics of the Internet. For example, the power-law characteristic is questioned [32, 92]. Although it does not affect the problem we discuss here, it may invalidate some of

the algorithms and analysis based on this assumption. Moreover, recently the work [119] also points out that seemingly disjoint layer-3 links might share a common layer-2 device. This will result in inaccuracies if inferred topologies are used to analyze the diversity of paths between hosts since the failure probability of two paths may not be independent.

Moreover, there are alternative methods available for network topology discovery. For example, the mrinfo tool [120] exploiting the IGMP protocol is proposed to discover the Internet topology. However, it cannot be directly used here since the problem discussed endeavours to find the connection relationship covered by the end hosts instead of all the connections an end host can obtain by recursively probing newly discovered routers.

Recently, there have also been efforts to infer the AS/router dual-level topology [121]. The accuracy of this type of topology depends on the targeted network and external BGP information is needed to verify its correctness. If its accuracy can be guaranteed, this dual-level topology might be of interest to those applications that aim to take AS-disjoint paths when participating hosts are scattered across multiple ASes.

6.4 Non-Router-Assisted RTD-Selective

In contrast to the router-assisted approaches, tomography-based topology inference does not need router cooperation. Therefore, this method is preferable especially when an increasing number of routers are configured so as not to respond to traceroute messages or even to discard them [103]. However, it usually requires the use of more carefully designed probes together with complex algorithms such as Maximum Likelihood Estimation (MLE) [122] due to the limited information obtained by the E2E probes. Therefore, research mainly focuses on reducing the probing overhead and proposing new algorithms offering high accuracy in inferring the topology based only on E2E probe information. In this section, the basics of tomography inference are firstly explained. This is followed by a discussion and analysis of the state-of-the-art in this field.

6.4.1 "Tomography" Basics

"Network tomography" was first introduced by Vardi [123] to obtain finer-level metrics from either passively or actively collected information. A typical example of passive tomography is to estimate the source-destination traffic matrix by exploiting the traffic information obtained from routers [123]. One instance of active tomography is to infer "internal" link-level delay and loss performance based on active E2E network measurements [124]. Nowadays, this term is expanded to describe the process of inferring the logical topology using multicast/unicast based methodologies [91]. They are termed Network Performance Tomography [125] and Network Logic Topology Tomography respectively, where the former usually assumes the network topology/routing information is readily available [124, 126].

In this chapter, we view Network Performance Tomography as schemes that nevertheless exploit Logical Topology Tomography and therefore focus our attention on the latter and use topology tomography for simplicity. Multicast-based tomography is usually employed to estimate the network performance [91]. Moreover, it typically requires all the hosts and routers involved to belong to common multicast groups in practical scenarios. Although it can acquire richer information, it is considered to be less practical than unicast-based schemes as many networks do not support multicasting at the IP layer [127]. Therefore, we focus on unicast-based methodologies here. We refer interested readers to the work of M. Coates [91, 124], L. Denby [62], E. Lawrence [128] for more details on Network Performance Tomography and multicast-based topology tomography. Recently, P. Sattari *et al* propose a combination of network tomography and network coding implemented at intermediate nodes [129] to infer the topology and internal link performance. However, it requires additional coding functionality to be present at the routers, which is yet to be widely adopted. As such we subsequently omit it.

Shared congestion estimation techniques [42, 130, 131] also exploit information obtained from the engineered E2E probes. However, one of the major differences from topology

tomography is that shared congestion estimation focuses on local and finer granularity such as the possibility of packet drop along the shared path [42]. Topology tomography, on the other hand, stresses the perspective of the whole network, i.e. the inference of the topology and/or metrics from all the constituent links if possible. To the best of our knowledge, this chapter is the first to provide a comprehensive review of the recent developments in topology tomography.

The basic procedure of logical topology inference using tomography is illustrated in Figure 6.7. A group of nodes (typically, one source and N destinations, i.e. a 1-by-N architecture) view the internal structure as a black box and send instrumented packets from the source to multiple destinations in order to obtain meaningful information. Afterwards, they exploit correlations between the sets of measured data to make an inference. In the final step, aggregation and/or other statistical means are used to estimate the potential connectivity among these nodes. Similar to RA methods, tomography-based inference can be classified into two types according to whether the probing and inference process are coupled or not. Usually, an iterative procedure [127, 132] can reduce the cost of network measurement when compared with the sequential ones [122]. In this section, we focus on: (1) the principles and assumptions behind topology tomography; (2) probe design. A detailed comparison of the inference algorithms will be presented in Section 6.4.2.

Generally, the following are assumed to be satisfied in tomography-based inference:

1. **Spatial independence**: Delay (and other additive metrics, such as loss rate) experienced by packets are independent along different links;

2. **Temporal independence**: Delay (and other additive metrics, such as loss rate) experienced by packets on the same link are Independent and Identically Distributed;

3. **Stationary**: The substrate topology is fixed during the observation period;

A Group of End Systems(1-by-N type)
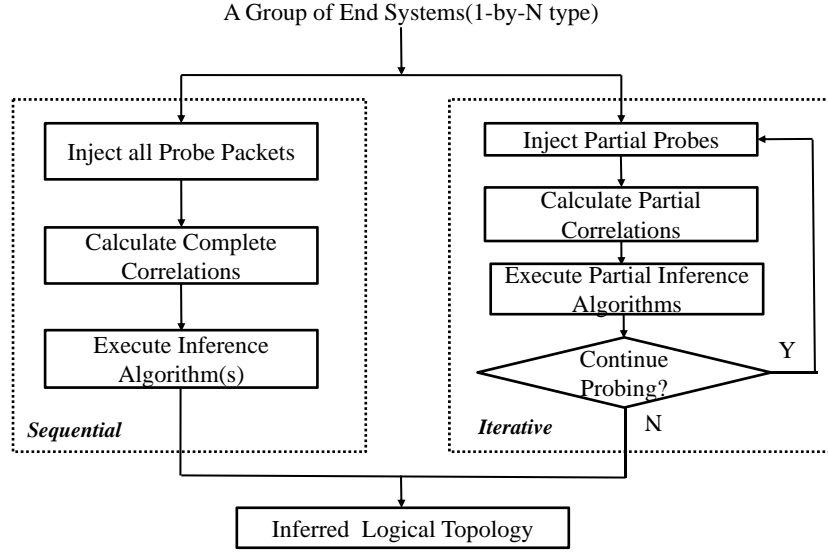
Figure 6.7: Workflows of the NRA RTD-Selective procedure

4. **Tree structure**: The routing topology from the source to a set of destinations is a (directed) tree and there is a single path from a given source to a destination.

The basic principle supporting logical topology tomography is that the more hops two paths share in common, the higher the correlation of the metrics (e.g. loss rate, delay) will be. Therefore, multiple pairs of correlation values can be exploited to infer the tree structure among multiple destinations. In order to obtain the correlation data effectively, several types of probe are designed. The earliest type is the "packet-pair" probe, these being two probes sent back-to-back from a source to two separate destinations, as shown in Figure 6.8 (a). This type of probe mimics the packet delivery behaviour of multicast-supported networks. So the two packets should experience similar network conditions if they are sent at a similar time toward the destinations. Parameters such as One Way Delay (OWD) or the loss rate of all 3-tuples (Source, Destination X, Destination Y) are measured. Consider the delay covariance as an example, the delay covariance measured for path (Host A, Host C) (i.e. P_{AC}) and path (Host A, Host B) (i.e. P_{AB}) will be larger than that of path (Host A, Host C) and path (Host A, Host D) (i.e. P_{AD}) in the network topology as depicted in Figure 6.1, according to this correlation principle, where Host A
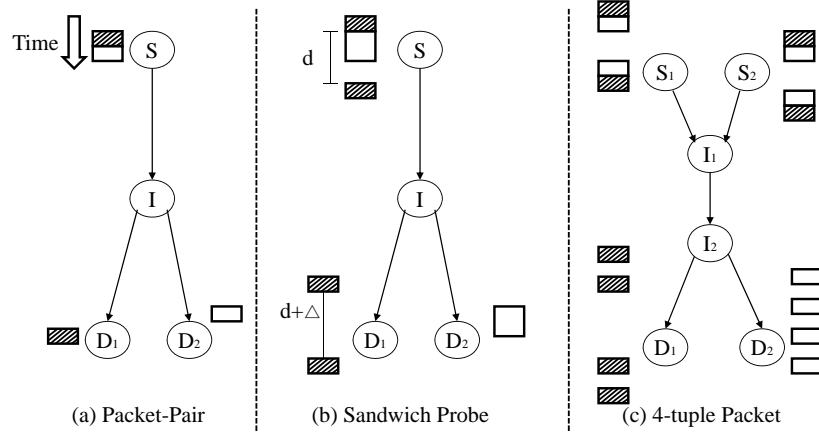
Figure 6.8: Probe design

is the source node and the rest are destination nodes. This can be expressed as:

$$Cov_{delay}(P_{AC}, P_{AB}) > Cov_{delay}(P_{AC}, P_{AD}) \tag{6.1}$$

where the delay can be substituted by other metrics such as loss rate. Then, based on the correlation matrix obtained, the internal nodes can be added sequentially or the whole topology can be determined depending on the algorithms employed.

However, one of the drawbacks of this probe approach is that synchronization is needed in order to obtain accurate OWD measurements for the destination node pair. Later, M. Coates *et al* developed the "sandwich" probe concept and chose the delay difference to obviate this requirement. As shown in Figure 6.8(b), two small packets separated by a large packet are sent out toward two destinations.

Recently, the "packet-pair" probe has been revised in order to be implemented in multiple-source tomography scenarios [133]. This probe is termed as 4-tuple packet and is issued from two sources as shown in Figure 6.8(c). All types of probes together with the metrics employed are summarized in Table 6.4.

Table 6.4: Types of tomography probe

Covariance Metrics	Probes	Working Scenarios	Special Requirement(s)
Delay	Packet Pair	Medium Load	Synchronization at receiver nodes
Loss Rate	Packet Pair	Heavy Load	—
Delay	Sandwich Probe	Light Load	The size of the large packet
Delay	4-Tuple Probe	Medium Load	Synchronization
RTT	TCP SYN Packet Pair	Medium Load	Uncorrelated paths from destinations to source

6.4.2 NRA RTD-Selective Algorithms

This section is devoted to summarizing algorithms proposed so far to infer the logical topology based on various kinds of available correlation information including delay difference, delay variance and loss rate and so forth. Tomography-based topology inference can also use multicast-based probes [134–136]. However, these approaches are limited to simulations of the proposed algorithms. In consequence we focus on unicast-based cases for practical reasons. Nevertheless, the discussion here is also applicable to analysis of data collected by sending multicast-based probes. Unless stated otherwise, the topology to be inferred is assumed to be a tree structure and the probes are issued at the single source node towards multiple destination nodes.

M. Coates *et al* [122] initiated the research into tomography-based logical topology inference using unicast-based probes. Sandwich probes are issued in the first step and the delay difference values are collected to obtain the correlation among all pairs of destinations. Then, a global optimization mechanism called Maximum Likelihood Tree (MLT) is proposed to find the most probable tree topology among the forest (i.e. the set of all possible trees for the given number of hosts). In order to reduce the search space of potential trees, a penalized MLT is proposed. Moreover, they also solve the tree search problem using a Monte Carlo Markov Chain (MCMC) method to reduce the

computational complexity. It includes two basic processes: birth (node addition) and death (node removal). The trees are transformed by either process and will be accepted according to the associated probabilities. The proposed method has several merits. One of the advantages is that it does not need clock synchronization among the destinations. Another is that it has comparatively high accuracy in its ability to find a global optimized tree. However, its computational complexity can be as high as $O(n!)$ and the overhead of using sandwich probes is $O(n^2)$ where n is the number of destinations. If deployed in a real network, it can take a long time and consumes considerable network resources.

Later, inspired by the Deterministic Binary Tree (DBT) algorithm proposed to infer the binary-tree topology in a bottom-up fashion by Duffield *et al* [137] using multicast-based logical topology tomography, M. Coates *et al* proposed another clustering-based algorithm called Agglomerative Likelihood Tree Inference (ALT) for use in unicast-based scenarios [138]. The proposed heuristic method takes into account the variability of measurement by considering its probability density function.

Afterwards, M. Shih *et al* [139, 140] formulated the logical topology inference problem as a Finite Mixture Model and proposed a hierarchical algorithm to infer the logical topology in a recursive manner. They discussed three correlation metrics: delay difference, delay covariance and loss rate and found that their effectiveness is sensitive to network-load. Different from previous methods, the basic process of this algorithm is to cluster the nodes according to their similarities using one of the metrics in a top-down fashion. After several iterations, every node will be separated into distinctive clustering groups and the whole process is then terminated. Also, the inferred tree topology does not necessarily conform to a binary structure. Furthermore, this approach has shown to be more accurate in terms of Edit Distance as compared to the DBT and ALT algorithms.

Then, Ni Jian *et al* [132, 141] analyzed how to obtain additive metrics based on multicast/unicast packet (packet pair for unicast) probes and formulated the problem as a Markov Random Field (MRF) model. They verified that additive metric correlation obtained at the terminal nodes can uniquely define a potential tree topology. Initially,

they proposed a Rooted Neighbor-Joining (RNJ) algorithm to infer the logical topology. The RNJ algorithm belongs to the family of clustering type algorithms. Furthermore, based on the observation that clustering type algorithms have shortcomings of poor probe scalability and the inability to support node dynamics, they further proposed a sequential algorithm. The proposed sequential algorithm can fuse information from various sources, such as incorporating traceroute and multicast/unicast tomography. Internet experiments on tree topologies showed that the combined methods can achieve high accuracy whilst incurring less overhead compared to the mechanism exploiting only one of these methods in isolation.

Recently, B. Eriksson *et al* expanded their earlier work [142] to take advantage of ordered destination sequence in tomography-based topology inference [127]. The basic observation is that if the destination nodes are arranged in a certain order, the number of probes needed to infer the topology between the source and destinations will be drastically reduced, as their correlation matrix will have a certain pattern. Therefore, instead of carrying out pair-wise probing and correlation calculations for $N(N-1)/2$ pairs of destinations, only calculations for N pairs is enough to infer the logical topology if the destination node is organized in Depth First Search (DFS) order. However, as the destinations may not necessarily be ordered in the desired sequence, the authors propose a method to obtain the sequence from the destination sets. According to their simulation results, the proposed DFS-ordered algorithm can decrease the probing traffic by 50% as compared to that of the previously most efficient sequential algorithm proposed by J. Ni *et al* [132].

Most of the work on topology tomography discussed above is focused on single-source multiple-destination topologies (i.e. 1-by-N topologies). Rabbits *et al* have led the research in multiple-source and multiple-destination topology inference (i.e. M-by-N structures) [133]. Theoretical analysis is provided to prove that M-by-N topology inference can be achieved by combining only 2-by-2 topologies. There are two types of 2-by-2 topology discussed: shared and non-shared. The shared 2-by-2 topology, by

definition, has a common path segment for the two source-destination pairs as depicted in Figure 6.8(c). The basic process of M-by-N topology discovery includes first discovering the 1-by-N tree topology for each source and then merging these topologies by inferring the relationship between the trees, i.e. whether two paths exhibits 2-by-2 structure by sending additional probes. More recently, Andrea et al [143] proposed an algorithm for merging multiple trees to obtain a complete topology. The proposed algorithm is based on sandwich probe and decision theory. Different from previous efforts, their proposed methods do not require additional probe traffic. Moreover, they also try to include non-branching routers by estimating the network diameter using TTL fields of the probes together with link capacity information.

In summary, a comparison is provided in Table 6.5. There are several issues worth mentioning about network tomography implementations. First, it is difficult to guarantee that the assumptions listed at the beginning of Section 6.4.1 hold true in real network scenarios. Noise will be introduced due to the inaccuracies in the topology inference process. In order to improve the accuracy, averaged metric values over multiple iterations are often adopted, which requires large amounts of traffic to be injected into the network. Second, as we can see from Table 6.5, most of the proposed algorithms are only focused on tree-like topologies. This limits their approach to specific situations. Moreover, the correlation metrics can only be applied in certain network scenario(s) as analyzed and proved by [140]. Therefore, the accuracy of the chosen metrics in practical situations depends on the characteristics of the networks involved.

6.5 Discussion and Summary

This chapter focuses on techniques addressing the issue of routing topology discovery among a group of participating hosts. A thorough discussion of various aspects of two main methodologies that can solve this problem is then presented, including their fundamental operation, a discussion on their main features of interest and a comparison

Table 6.5: Summary of tomography inference algorithms

(1) 1 or M source nodes, N destination nodes
(2) l-ary tree with depth of $O(log_l N)$ and $p(l)$ is sublinear in l
(3) Comput. Compl. (Computational Complexity)
(4) Pro. Compl. (Probing Complexity)

Algorithms	Topo. Type	Description	Comput. Compl.	Pro. Compl.[13]
MLT(2002)	Tree	Global optimization; Finding a most likely tree in the candidate forest.	—	$O(N^2)$
ALT(2003)	Tree	Clustering in a bottom-up fashion incorporating measurement probability density function.	—	$O(N^2)$
HTE(2005)	Tree	Clustering in a top-down fashion	—	$O(N^2)$
M-by-N tree(2005)	Mesh	Merging 1-by-N trees using 2-by-2 structure information	—	$O(M^2 N^2)^{note14}$
RNJ(2006)	Tree	Joining a node into the tree in a recursive manner by finding its neighbours.	$O(N^2 log N)$	$O(N^2)$
Sequential (2006)	Tree	A general framework merging information from different sources, such as traceroute and tomography.	$O(Nl log_l N)$	$O(Nl log_l N)$
DFS Ordering (2010)	Tree	Rearranging the destination group in a certain order and joining them sequentially in DFS order into the inferred topology using the covariance metrics.	—	$O(p(l)N log_l N)$

across the state-of-the-art research.

According to analysis throughout this review, none of the methodologies can produce

an exact routing topology among a group of hosts and it is difficult to measure the accuracy of the inferred topology with the true topology. Therefore, it is meaningful to compare the robustness of the proposed algorithms and techniques that take advantage of the inferred topology information given imperfect topological information.

As seen from our review, router-assisted methods still have practicality in real Internet applications for several reasons. Firstly, router-assisted methods are straightforward and simple to implement. Secondly, a high percentage of routers still reply to traceroute messages. Although a moderate amount of work has addressed the problem of reducing probe traffic or designing time-efficient AR or IP Alias resolution strategies, there is no work to-date that considers all the problems as a whole. Since the issues associated with the traceroute-based methods are related and there is an increasing trend for routers to be configured to operate as ARs, it is worthwhile and practical to investigate some or all of these problems together.

In contrast to RA methods, tomography-based approaches have the benefit of not relying on any form of router cooperation and are gaining momentum for topology inference solely based on E2E probe information. Nevertheless, there are several aspects that still need extensive research before a practical implementation of network tomography for medium/large network topologies can be introduced. Firstly, tomography-based algorithms involve injecting a large amount of extra traffic into the network as well as prohibitive computation times. Therefore, simplifying the operation whilst maintaining accuracy when applied to large networks remains an active area of research. One means of reducing the complexity would be to combine information from other sources such as traceroute, partial routing table data and so forth. Secondly, to-date, the research has been focused on small/medium-sized controlled network environments. The effectiveness of tomography inference for medium/large topology scenarios still needs to be evaluated. Finally, most of the state-of-the-art work remains focused on tree-like topologies. How

---

[13]In order to mitigate the noise introduced in the variance calculation, a repetition time is usually adopted, such as obtaining metrics using 1000 packet pairs for each destination pair. The main factor considered here is the relationship of probing complexity with the number of source/destination nodes.

[14]Assuming this algorithm uses packet pair probes.

this can be generalized to estimate the topology among a group of nodes similar to that achievable with RA methods would be of great value to promote its adoption more generally.

Chapter 7

Impact of Substrate Network Topology Availability and its Accuracy

7.1 Overview

In this chapter, we focus on the scenarios where substrate network topology information is not readily available to provider-independent overlay networks and strategies are required to obtain it. More specifically, topology inference among a group of hosts needs to be carried out. The impact of the inferred topological information availability and its accuracy on provider-independent overlays as compared to that of the true topology is investigated. We first describe the problem and the network model adopted here. Then, we present simulation results of the impact of inaccuracies in the inferred substrate topology information.

7.2 Problem Description

In this chapter, it is assumed that the target topology is a router-level topology among a selective group of hosts which usually resides in stub domains. As discussed in Chapter 6, there are two kinds of active probing strategies available, namely, traceroute-based and tomography-based. We focus here on traceroute-based methodologies since tomography-based methods cannot be used to address this issue. There are three main steps to obtain such a router topology, namely: (1) Probe injection; (2) Initial probe data processing (This includes private IP addresses removal, traceroute information verification and IP Alias Resolution and so forth.) and (3) Anonymous Router resolution. Obviously, there are many factors in this process that can result in an inaccurate topological inference. We refer to these factors as "distortion factors" for brevity and explain each one in more detail.

Firstly, the classic traceroute tool is not adequate to cope with the anomalies found in the Internet. Therefore, if it is employed, two problems can arise: the inability to discover the true nodes/links and the false reporting of links [102]. In order to reduce the impact of this distortion factor, Paris traceroute has been proposed and is reported to work more satisfactorily in terms of avoiding anomalies such as loops, diamonds and cycles [101].

The second distortion factor originates from the fact that some routers will be missing if probe overhead reduction is of high priority. As discussed in the work [44], given the assumption that no measurement noise such as anonymous router or router aliasing is considered, more than 90% of routers can be discovered with each host probing strategically chosen destination nodes. However, if complete topological information (including node and link) is required, then there is little scope to reduce the probe overhead.

The third and fourth distortion factors are the result of two practical features of the Internet. The first is that one single router may have multiple IP addresses assigned to each of its interfaces (i.e. IP alias). Therefore, in the aggregated traced paths, multiple
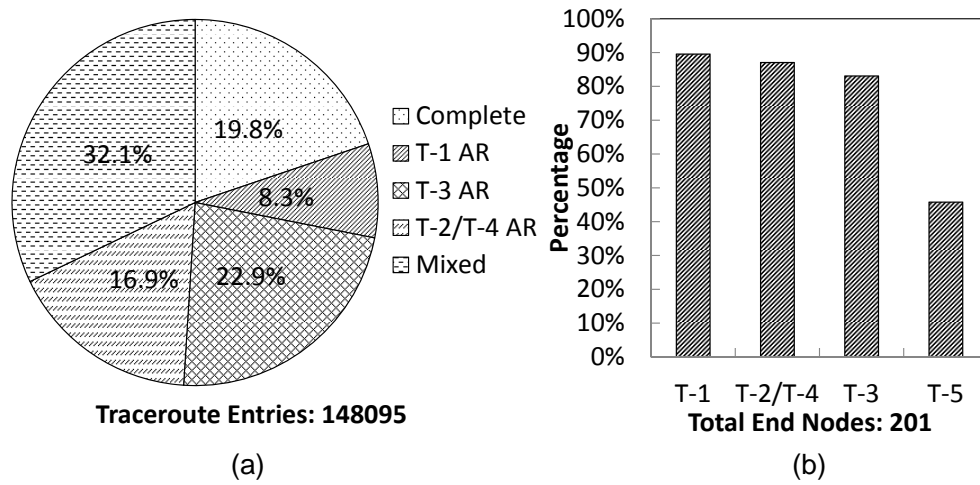
Figure 7.1: Anonymous router presence analysis using a real dataset: (a) analysis based on traceroute entries; (b) analysis based on end-nodes

nodes actually belong to a single router. The techniques that address this issue are termed IP Alias resolution. The other is that routers can be configured not to respond to probe packets and thus cause a great inflation in the number of nodes and links in the aggregated topology graph. The techniques that address this issue are termed as Anonymous Router (AR) resolution.

There is a moderate amount of literature endeavouring to reduce the impact of each distortion factor as presented in detail in Chapter 6. We analyze the frequency of AR with different behaviours present in the traceroute entries using the dataset provided by the iPlane Project, including 201 Planetlab nodes tracerouting each other. As shown in Figure 7.1[1], the AR behaviour is common in the real network. From all the traceroute entries we have obtained, no more than 20% of the entries have a complete traceroute path (i.e. without any AR issue). Thus, additional strategies are needed to tackle this if a more accurate topology is preferred. Moreover, an accurate topology cannot be guaranteed to be obtained as formally proved by [115] even if there is only a moderate amount of AR nodes in the network.

---

[1]T-5 anonymous router (private IP address) is not counted when checking each tracroute entry since it is viewed as having an IP address. Moreover, later in our discussion, T-5 is treated as an independent node in the inferred topology.

In our work, we mainly focus on the last two factors, namely IP alias resolution and AR resolution. Moreover, we use pairwise tracerouting information in our later simulations and analysis, thus the second factor is avoided. Although previous works [43, 48] as well as our own work, described in Chapter 5 and 6, have shown that substrate network information is important to help improve overlay performance, it is assumed that this information is readily available. To the best of our knowledge, this is the first attempt to address how the availability and accuracy of the inferred topology can affect the performance of provider-independent overlays.

7.3   Network Model

7.3.1   Notations and Definitions

The notations adopted are similar to that in [104] and they are explained as below:

- $G(V, E)$: A router-level network graph where V represents the set of vertices and E represents the set of edges connecting two vertices. $G_i(V_i, E_i)$ represents the inferred topology and it usually has different number of nodes and links as compared to the true topology.

- $P(v_i, v_j)$: It represents a sequence of vertices connecting from v_i to v_j. The path can be chosen based on specific criteria such as minimum delay, shortest hops and so forth. $P(v_i, v_j)$ is treated as different from $P(v_j, v_i)$ due to path asymmetry reported in the Internet [144].

- v^*: An anonymous router that does not behave as specified by ICMP.

- $Trc(v_i, v_j)$: It is a function of $P(v_i, v_j)$ where the trace visits each vertex $v_k \in P(v_i, v_j)$ originating from v_i, traversing the intermediate nodes, to v_j and return a list of nodes including the source and destination identifiers as the output. In the ideal scenario with no ARs, $Trc(v_i, v_j) = P(v_i, v_j)$. If there is one or more

anonymous routers present in the tracerouted path, v^* will be used to denote it in the trace.

- $S_*(v_i, v_j, N_*)$: This denotes the a string that has v_i and v_j as known end nodes and N_* specifies the number of anonymous nodes this string possesses.

7.3.2 Resolution Techniques

For IP alias resolution, we adopt the analytical IP alias resolution method proposed in [145] because it is reported to be more efficient in terms of resolving IP aliases without additional probe traffic and we meet the basic requirement of this method (i.e. both the forwarding and reverse paths between two end hosts is essential for this technique). The basic principle of this method is that /30 or /31 subnet IP addresses are usually assigned to a point-to-point link. Therefore, by comparing the forwarding and reverse paths between an end-node pair, i.e. $Trc(v_i, v_j)$ and $Trc(v_j, v_i)$, the IP alias pair can be identified. An example is presented in Figure 7.2 [145].

As for AR resolution, there are several techniques available as summarized in Chapter 6. Here, we focus on those techniques that are more practical with low computational
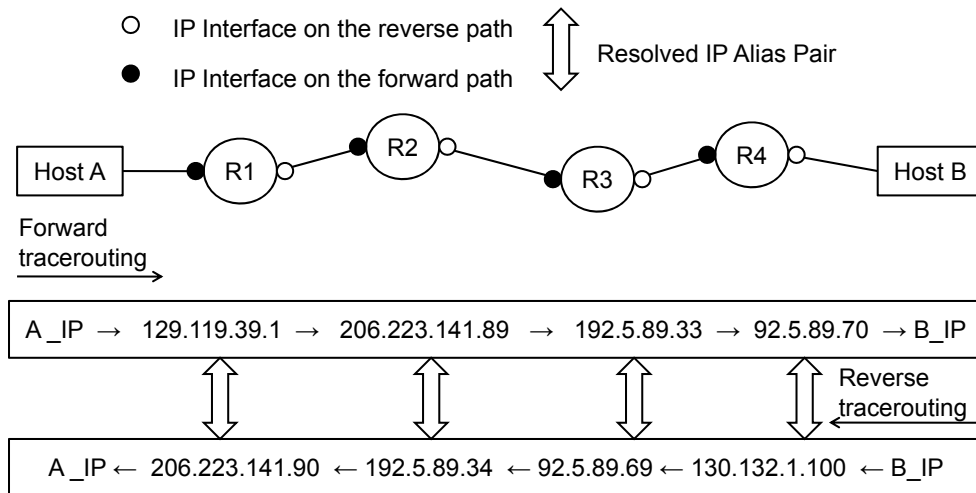
Figure 7.2: An example of IP alias resolution

complexity. They are Initial Pruning (IP), Neighbouring Matching (NM) and Graph-Based Induction (GBI) methods and their principles are explained briefly below:

- **IP**: The principle of this method is simple, it gathers all the $S_*(v_i, v_j, N_*)$ and allocates a unique ID to all these anonymous nodes. Then, instead of assigning a unique ID for each anonymous node in all the traceroute entries, the ARs are assigned by looking up the IDs in the $\{S_*(v_i, v_j, N_*)\}$ set. As indicated by an analysis using a real network dataset in [104], this method reduces the largest amount of AR nodes as compared to other procedures considered.

- **NM**: This approach applies the constraint of the IP method less stringently when assigning a unique ID to AR nodes. Two AR nodes are given the same ID as long as they have one common known end-node and do not appear in the same traceroute entry. Obviously, this can result in merging ARs that should not be merged since this merging criterion is not generally true in the real network.

- **GBI**: This method is shown to be the most efficient so far in terms of edit distance (in obtaining a true topology) and it resolves AR nodes by identifying certain structures (i.e. star, bipartite) in the aggregated traceroutes (details are provided in Chapter 6). The GBI method includes mechanisms similar to the IP and NM methods.

According to the explanation of these three methods, it is conjectured that the last two methods will incur false merging of two AR nodes in the resulting topology and none of these methods can guarantee to obtain an accurate topology. However, in this work, we have no intention to obtain the true substrate topology and only endeavour to determine the information needed for overlay network use, be it delay or overlap number of two paths in the substrate network.

7.4   Performance Evaluation

As discussed in Chapter 5, there is little difference in terms of resilience among various topology construction algorithms considered given an appropriate overlay setting even if substrate topology information is exploited. Therefore, we only focus on analyzing the performance of the ROMCA overlay in terms of providing application mapping service.

We first exploit a real-network traceroute dataset. Since it is not possible to obtain the true topology for the real network dataset, we also use controlled simulations exploiting a synthetic topology in order to further compare the effectiveness of various AR resolution techniques.

7.4.1   Evaluation with a Real Network Topology

The pairwise tracerouting dataset provided by iPlane project is used. After pruning incomplete traceroute entries, 95 out of 201 nodes are selected for overlay mapping use. Both the IP alias and the anonymous router issues are present. However, no techniques can obtain the true underlying topology among these end-nodes. Hence, we use the inferred topology obtained after IP alias resolution and AR resolution are undertaken using the GBI method to obtain a near true topology because it is reported to be the approach that can obtain the topology closest to the true topology for comparison purposes. There are five different types of inferred substrate topology and they are listed as follows:

1. *Raw*: This topology is an aggregation of the traceroute entries without any IP alias and AR resolution processing;

2. *Alias*: This topology is obtained after resolving IP aliases.

3. *IP*: This topology is obtained after resolving IP aliases and implementing AR resolution exploiting the *IP* method;

4. NM: This topology is obtained after resolving IP aliases and implementing AR resolution exploiting the NM method;

5. GBI: (aka $True$) This topology is obtained after resolving IP aliases and implementing AR resolution exploiting the GBI method;

7.4.1.1 Simulation Settings

The application mapping obtained by the substrate-topology-aware $pQoSMap$ heuristic with these different topology information sets are notated as Raw, $Alias$, IP, NM and $True$ accordingly. For the application request, five topologies [2] with full mesh, 50% connectivity, 25% connectivity, ring and tree are considered. The overlay node number varies from 20 to 40 with an interval of 5. The maximum overlay delay requirement is set so that the remaining connectivity for the simplified overlay topology (referred to as remaining connectivity later for brevity) varies from 50% to 100%.

For comparison purposes, the existing heuristic $QoSMap$ which is oblivious to substrate topology is also included in the analysis. The metrics we focus on in this chapter include D_{avg}, O_{wb} and C_r and they are formally defined in Chapter 5. Only typical results are presented since the remaining results are comparable. An explanation of the simulation platforms and a verification of their correctness is provided in Appendix B.

7.4.1.2 Results and Analysis

We first present the results for the scenarios with 40 application nodes and 90% remaining connectivity in the simplified overlay network and they are shown in Figure 7.3 and Figure 7.4. As depicted in these two figures, the proposed heuristic can work much better in terms of securing effective resilience. For example, with a full mesh topology

---

[2]Please note that the performance difference given different application topologies cannot be compared because they have different features (i.e. different number of links). The proposed heuristic in Chapter 5 endeavors to achieve enhanced QoS and effective resilience irrespective of the given application topology. We use a variety of application topologies in order to evaluate the heuristic across a range of scenarios.
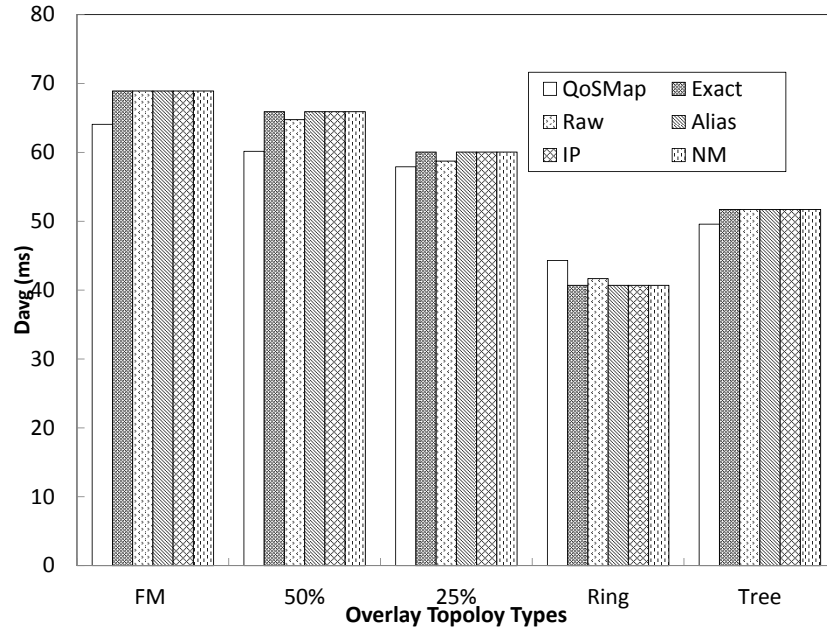
Figure 7.3: D_{avg} evaluation in the scenarios with 40 application nodes and 90% remaining connectivity
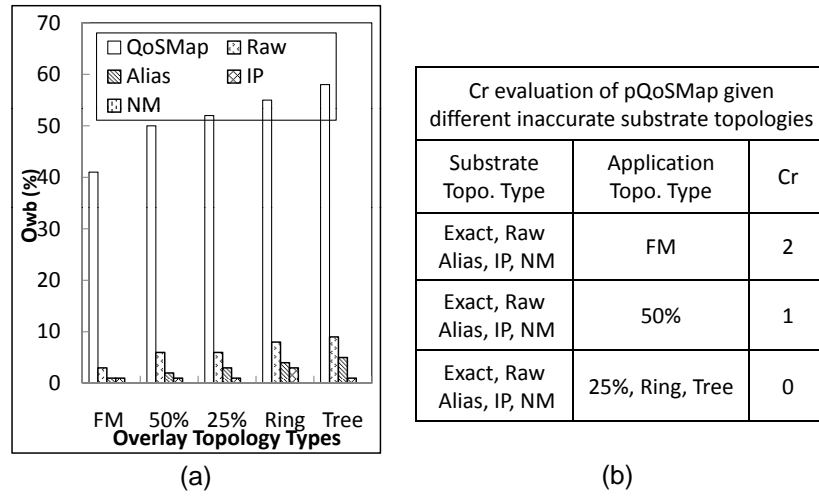
Figure 7.4: O_{wb} and C_r evaluation for the same scenarios as Figure 7.3 (Note: only non-zero values are shown here.)

request, there is about 40% overlap between the working and backup paths in the solution provided by the *QoSMap* method whilst *pQoSMap* can achieve less than 5% overlap even given only raw substrate topology information. This is achieved at the expense of a slight higher average delay value and a small number of extra overlay nodes to achieve the diversified backup paths.
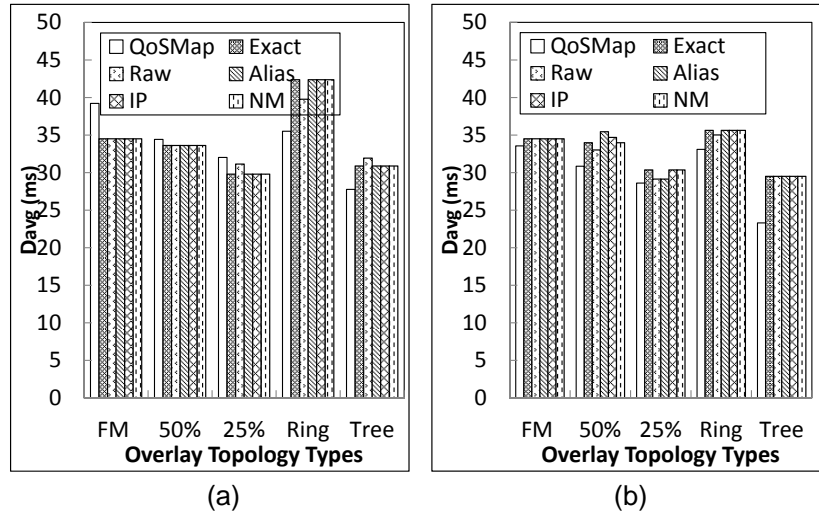
Figure 7.5: D_{avg} evaluation in the scenarios with 25 application nodes and two different remaining connectivities: (a) 80%; (b) 60%
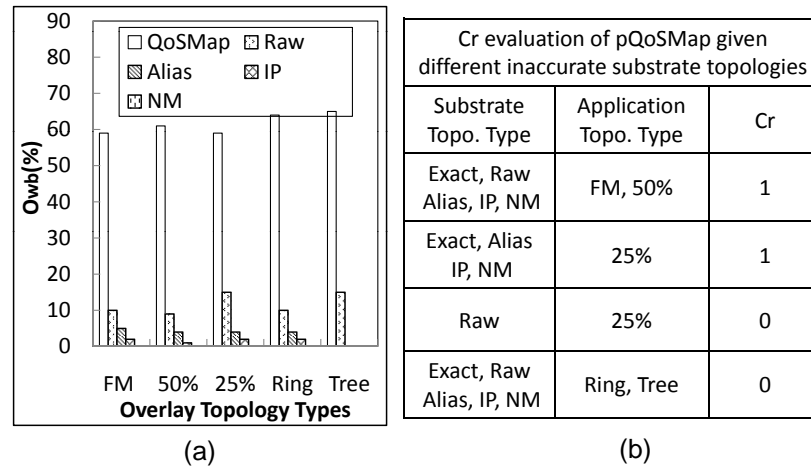
Figure 7.6: O_{wb} and C_r evaluation in the scenarios with 25 application nodes and 80% remaining connectivity (Note: only non-zero values are shown here.)

Moreover, IP alias and AR resolution techniques are instrumental in terms of further reducing the overlap value obtained in the mapping solution as shown in Figure 7.4. To be more specific, the NM method performs the same as the true topology (aka the GBI method) and the IP method comes second. Similar conclusions can also be obtained from the scenarios with 25 overlay nodes and 80% and 60% remaining connectivities in the simplified overlay topology. These results are shown in Figure 7.5, Figure 7.6 and Figure 7.7.

Figure 7.7: O_{wb} and C_r evaluation in the scenarios with 25 application nodes and 60% remaining connectivity (Note: only non-zero values are shown in this graph.)

7.4.2 Evaluation with a Synthetic Topology

Since the true topology cannot be obtained with the real-network dataset, we have further used simulations with a synthetic topology to verify the effectiveness of various AR resolution techniques.

7.4.2.1 Simulation Settings

A synthetic topology with 3200 nodes and 20000 links [44] [3] generated using the GT-ITM topology generator is adopted as the substrate layer. For this topology, we focus on comparing the effectiveness of different AR resolution algorithms in reducing the O_{wb} value of the *pQoSMap* heuristic given inferred topology information [4]. This is possible because we have the genuine topology information and it is denoted as *True*.

For comparison purposes, the existing heuristic *QoSMap*, which is oblivious to substrate topology, is also included in the analysis. There are five types of inferred substrate

---

[3] The topology generated is a two-layer hierarchy of transit networks (with 8 transit domains, each with 16 randomly-distributed routers) and stub networks (with 256 domains, each with 12 randomly distributed routers) [44].

[4] There is no IP alias issue since each node has a unique ID.

topology information, namely, $True$, Raw, IP, NM and GBI. The application mapping solutions obtained by the $pQoSMap$ heuristic with these topology information are denoted as $True$, Raw, IP, NM and GBI respectively.

We randomly choose 80 nodes in the stub areas to be candidates for application mapping[5]. The percentage of AR nodes in the substrate network is 10% (In all these AR nodes, 90% are T-1 and the rest are T-3.) and it is assumed that the candidate overlay nodes are fixed. As for the application request, the topology and delay is set similar to the simulation with the real-network topology and the overlay node number varies from 10 to 30 with an interval of 5.

7.4.2.2   Results and Analysis

As shown in Figure 7.8 and Figure 7.9, although the $QoSMap$ heuristic can provide slightly better average delay performance in most cases, there is huge percentage of overlap in the mapped working and backup paths. On the other hand, the $pQoSMap$ heuristic can ensure no overlap in the mapped solution whilst meeting the delay requirement but is inferior to that of the $QoSMap$ heuristic due to the diversity constraint. However, this comes at the expense of a higher number of substrate nodes included in the overlay mapping solution for backup purposes. This is again supportive of the conclusions we have summarized in Chapter 5 but with a more realistic synthetic topology.

In the rest of this section, we focus on the impact of the $pQoSMap$ heuristic given various inferred topologies. Firstly, as shown in Figure 7.8, the D_{avg} values obtained by the heuristic exploiting the four inferred topologies are similar to each other and they are very close to the D_{avg} values obtained given the true substrate topology information. Secondly, as depicted in Figure 7.9(a), we can conclude that accurate substrate topology information is essential for securing a truly resilient overlay mapping solution. Even if only the raw topology information is exploited, the percentage of working and backup

---

[5]Verification of this randomness and the simulation platform is provided in Appendix B.
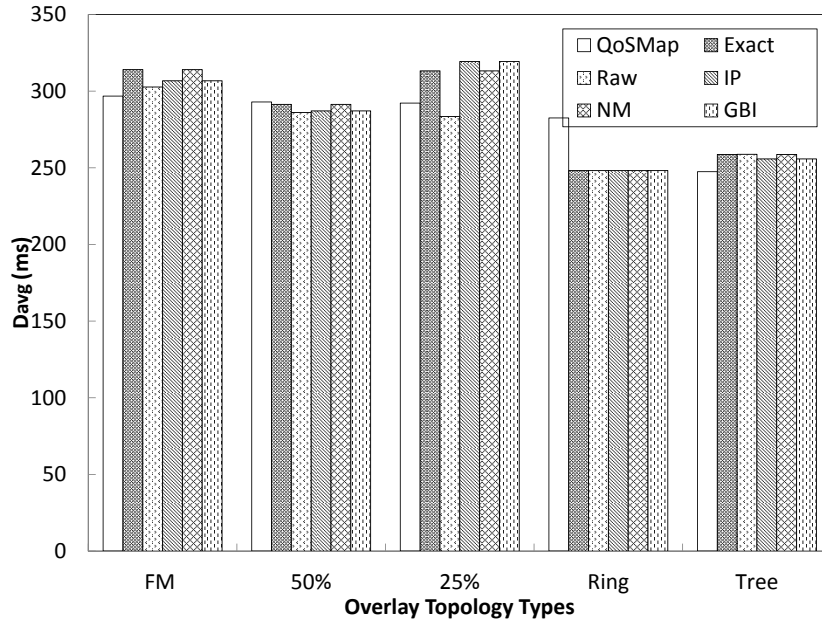
Figure 7.8: D_{avg} evaluation in the scenarios with 30 application nodes and 90% remaining connectivity

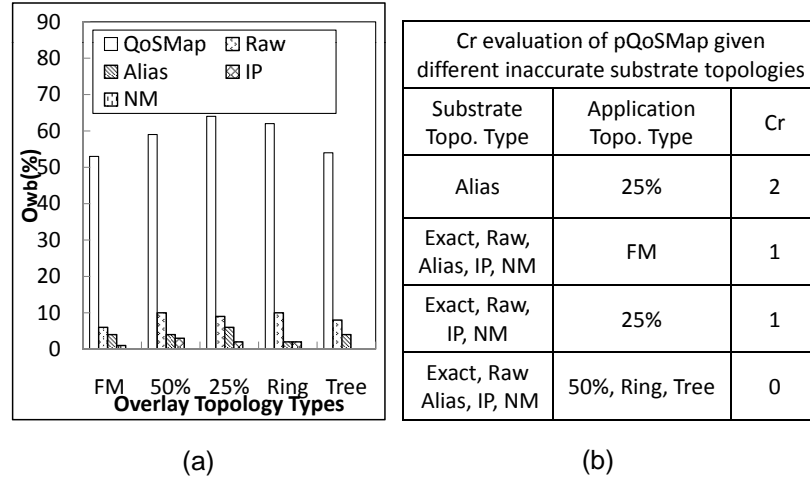Figure 7.9: O_{wb} and C_r evaluation with 30 nodes for the application request and 90% remaining connectivity (Note: only non-zero values are shown here.)

path overlapping can be reduced by as much as 45%. This is because even if overlap is detected between two incomplete paths, they will not be chosen as the working and backup paths for a single application link mapping.

We further analyze the effectiveness of different AR resolution techniques in helping reducing the O_{wb} value. As the results of the $pQoSMap$ heuristic for various substrate topologies shown in Figure 7.10, AR resolution techniques can help increase the effective resilience of the overlay mapping solution. For the three techniques we evaluate, the NM method is the most effective (i.e. resulting in no overlap in the mapped solution). This is different from the evaluation in [104] where the NM method is considered worse than the proposed GBI method in terms of obtaining a topology closer to the true topology. The reason lies in the fact inaccurate overlap information does not affect the overlay mapping significantly unless two truly overlapped paths are considered not overlapped (termed as false positive (FP)). Although the NM method mistakenly merges anonymous nodes (i.e. resulting in overlapping values bigger than the actual value) as compared to the GBI method, it has a lower FP value as compared to the GBI method. Therefore, the NM method can help prevent the proposed heuristic choosing uncertain paths and thus reduces the O_{wb} value of the overlay mapping solution.

Although in current simulation settings, the NM method can also obtain a zero-
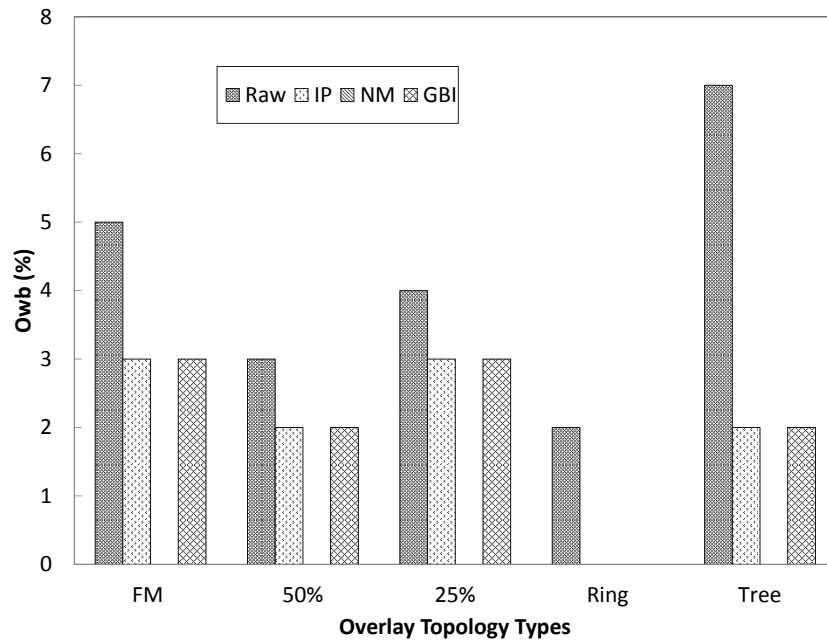
Figure 7.10: O_{wb} evaluation for $pQoSMap$ with different inferred substrate information for the same scenarios as in Figure 7.9
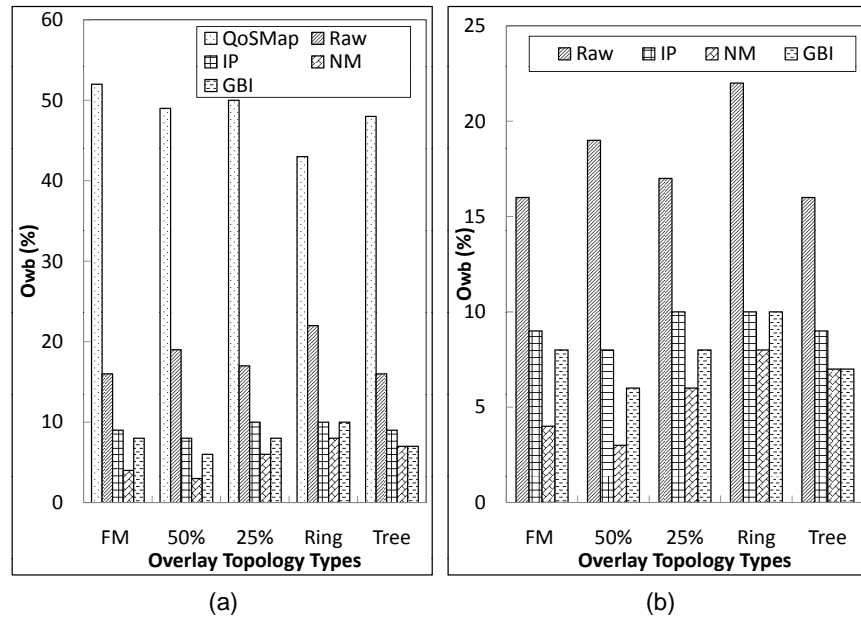
Figure 7.11: O_{wb} evaluation with 30 nodes for the application request and 90% remaining connectivity and 20% AR ratio: (a) *QoSMap* and *pQoSMap*; (b) *pQoSMap* with various inferred substrate topologies

overlap value, it is conjectured that this would not always be the case since the NM method cannot secure to obtain the true topology. Thus, in order to verify this conjecture, we evaluate using a higher AR ratio of 20%[6]. Figure 7.11 presents the O_{wb} value for this setting. We also define the following notations for further analysis from the perspective of path overlap information. They are **Equal, Bigger, Smaller** [7] and **FP**. The first three notations mean the extent to which the paths overlap as measured from the inferred substrate topology is the same or bigger or smaller than that of the true topology. FP denotes that a non-zero path overlap value in the true topology is mistakenly measured as 0 in the inferred topology. So, the inferred topology with least amount of FP entries should be the one helping the most in terms of reducing the O_{wb} value in the mapping solution. As shown in Figure 7.12, the topology inferred with the NM method has the lowest FP value among the AR resolution techniques considered.

The analysis presented above is still valid for an application request with other num-

---

[6]This setting is considered only for verifying this point and not used in any other results presented here.

[7]It should be a non-zero value.
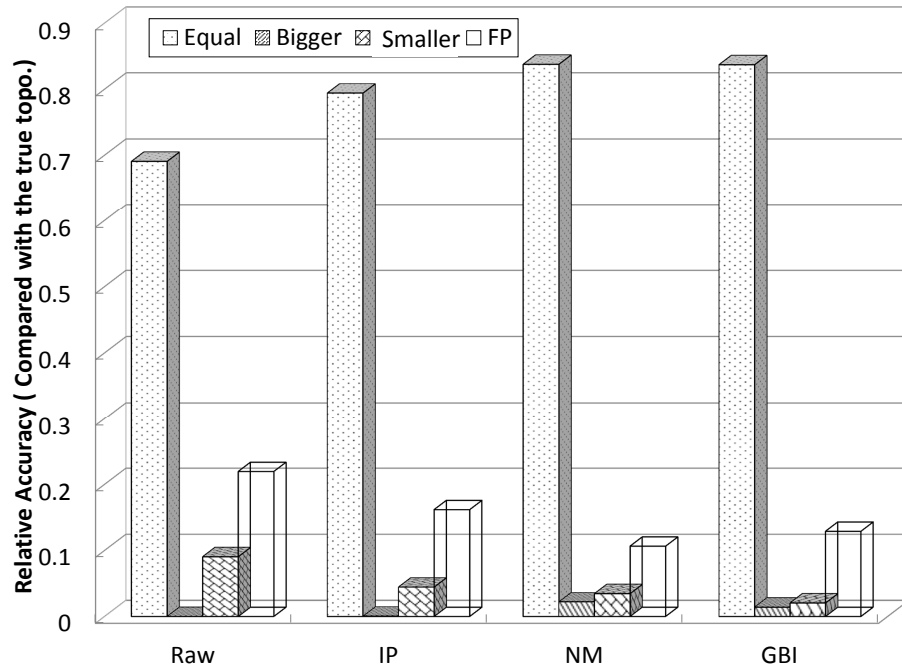
Figure 7.12: Overlap value accuracy of various inferred topologies as compared to that of the true topology with 30 nodes in the overlay and 20% AR ratio in the substrate layer
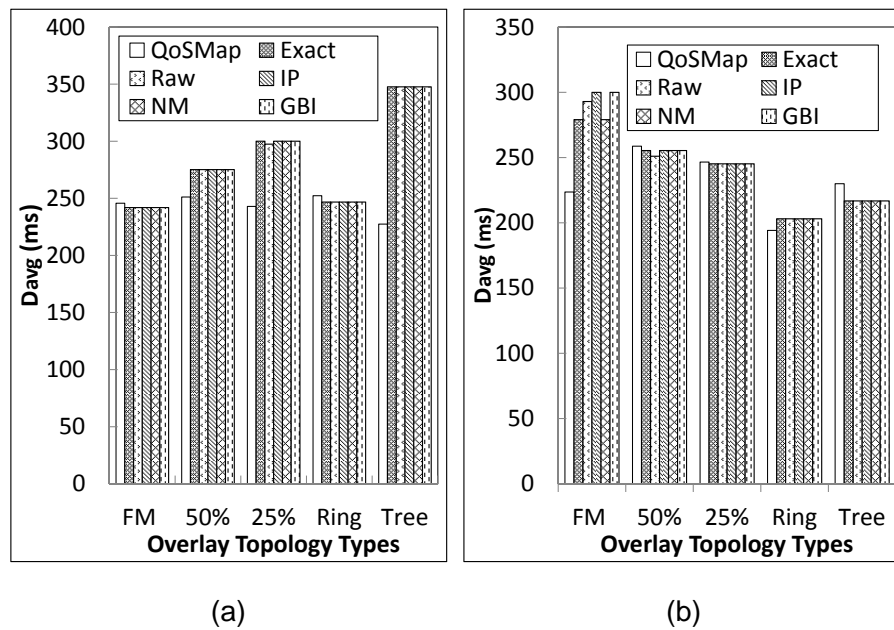
Figure 7.13: D_{avg} evaluation in scenarios with 15 nodes for the application request with two remaining connectivities: (a) 100%; (b) 70%

(a)   (b)
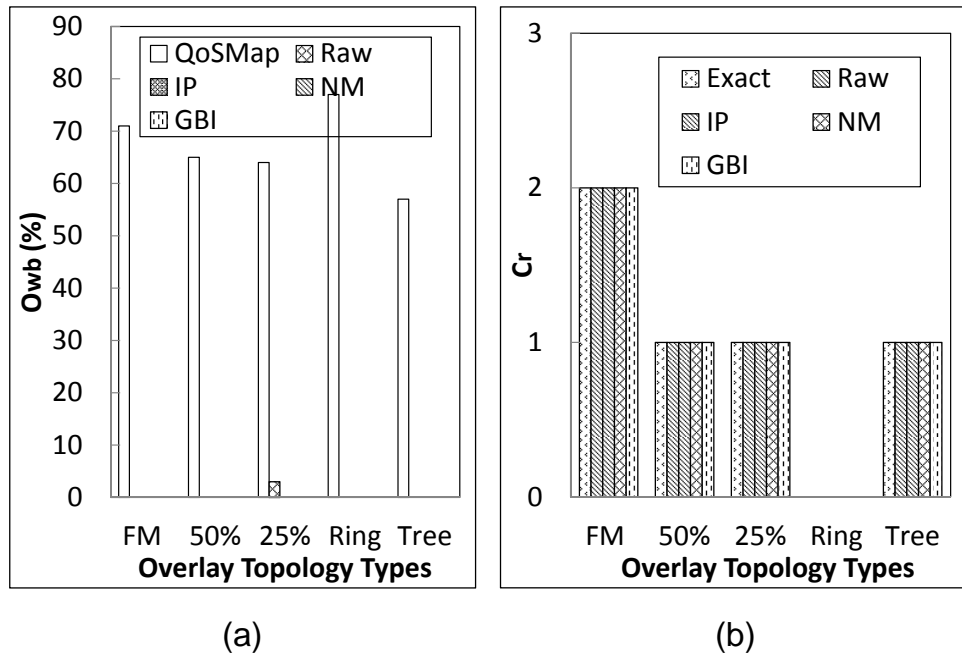
Figure 7.14: C_r and O_{wb} evaluation with 15 application nodes and 100% remaining connectivity (Note: only non-zero values are shown in this graph.)
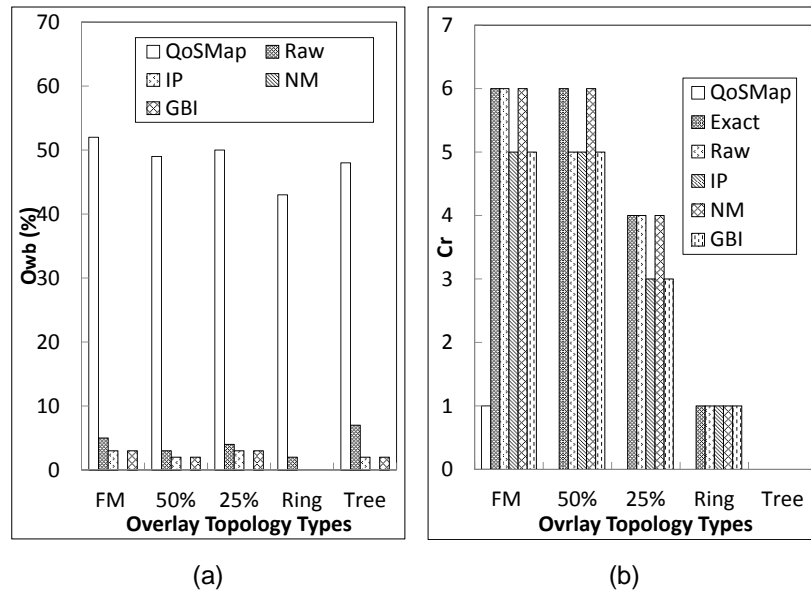

(a)   (b)
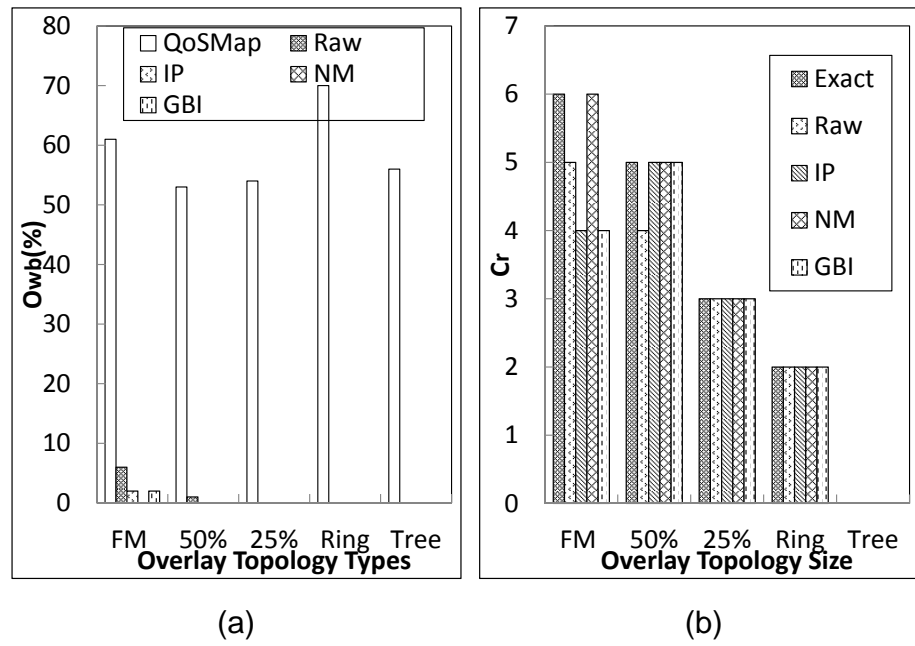
Figure 7.15: C_r and O_{wb} evaluation with 15 nodes for the application request and 70% remaining connectivity (Note: only non-zero values are shown in this graph.)

bers of nodes and the results are presented in Figure 7.13, Figure 7.14 and Figure 7.15. As shown in these graphs, although the raw topology information is inaccurate and incomplete, it can drastically reduce the possibility of overlap in the mapping solution. Moreover, the use of AR resolution techniques are able to further reduce the percentage of overlap. Among all the AR resolution methods discussed, the NM method performs the best.

7.5   Summary

In this chapter, we first discussed the problem of inferring an accurate substrate topology for the nodes under the administration of the ROMCA overlay. We focus the discussion on two of the issues, namely, the presence of IP aliases and anonymous routers in the traceroute entries. Then, we investigate the impact of substrate topology availability and accuracy on the performance of provider-independent overlays for obtaining an application mapping with enhanced resilience and QoS in both real-network and synthetic topologies.

As shown in the analysis, even raw substrate network topology information can help improve the effectiveness of the proposed heuristic in providing effective resilience. Furthermore, various IP alias and AR resolution techniques can help increase the resilience performance of the proposed heuristic. According to our simulations with a synthetic topology, the neighbouring match AR resolution technique is the most effective method in finding a mapping solution with lower overlap between the working and backup paths.

Chapter 8

Conclusions and Future Work

8.1 Overview

In this thesis, we have addressed various issue of provider-independent overlay networks. Firstly, we proposed a new provider-independent overlay architecture which assumes little support from ISPs. Then, based on this framework, we focused on providing two different services, namely, overlay topology construction for providing resilience service and application mapping exploiting the overlay. Moreover, we also investigated the impact of substrate network information availability and accuracy on the performance of the overlay. In this chapter, we summarize the key findings of our work and point out potential future work that could be undertaken based on our work.

8.2 Research Summary

8.2.1 ROMCA Architecture

In this thesis, we have proposed a new overlay architecture named Resilient Overlay for Mission-Critical Applications (ROMCA). One of the main characteristics of the proposed

overlay is that it assumes little support from network service providers. Although sitting overlay nodes upon routers is better in terms of providing additional resilience and QoS, the inter-trust issues between ISPs must be resolved. ROMCA offers a transient solution to provide resilience to end-users in multi-domain environment without impacting the current operation of ISPs. Moreover, ROMCA exploits a hybrid method of fulfilling resilient service delivery by organizing the overlay resources centrally whilst monitoring and routing in a distributed manner. This can reduce the burden of each overlay node obtaining complete substrate network information directly and allows them to cooperate with each other to fulfill the two services we focus on.

8.2.2 Overlay Construction for Providing Resilience Service

We have presented a simulation-based study of overlay topology construction for providing resilience services. Given that most previous work has focused on network-provider-dependent overlay topologies, we instead focused on provider-independent networks assuming full knowledge of the substrate network topological information. We have formulated the problem mathematically and proved its NP-hardness. Then, three heuristic approaches have been proposed to construct a highly resilient overlay.

The main findings of this study are:

- The proposed Least-Overlap Mapping of Regular Graph algorithm (LO-MARG) performs the best in all scenarios with AS-level topologies and it performs much better than other methods given a low overlay degree constraint. However, it possesses very high computational complexity since a large number of iterations are needed to search for a more resilient solution.

- The proposed Enhanced Dual-Layer-aware K-Minimum Spanning Tree algorithm (EDL-KMST) typically performs second best in terms of resilience among all the methods compared under the same overlay node degree constraint with AS-level topologies and performs the best with router-level topologies compared to the

methods considered. This EDL-KMST algorithm has much lower complexity. However, it cannot guarantee that the constructed overlay topology is a regular graph.

- Through extensive simulations with both AS-level and router-level topologies, it can be concluded that a random mapping of a regular graph can perform satisfactorily with an appropriate overlay node degree setting. The advantage of this method is that it does not need substrate topology information. However, if an overlay with guaranteed higher resilience is desirable, the two proposed substrate-topology-aware schemes are recommended.

8.2.3 Application Mapping to Achieve Enhanced Resilience and QoS

In terms of undertaking application mapping with enhanced resilience and QoS performance for the ROMCA overlay, we have proposed a novel overlay mapping model using an Integer Linear Program. Moreover, a heuristic has been proposed to solve the problem in a time-efficient manner with larger networks.

We have examined the performance of the two proposed methods through extensive simulations and the main conclusions of this study are summarized as follows:

- The proposed enhanced ILP model can provide much better QoS as compared to the heuristic solutions. Moreover, it can provide effective resilience with little additional overlay node resources. Although it is infeasible for use with larger networks, it still provides a useful benchmark for evaluating existing and new heuristic algorithms with small networks.

- The proposed heuristic can provide more effective backup paths compared to the state-of-the-art best solution. However, it does not necessarily provide the best quality of service. Moreover, a higher amount of additional substrate nodes may be needed for effective backup purposes as compared to the ILP solution. Nevertheless, the heuristic has very low computational complexity as compared to the

optimal solution.

8.2.4 Impact of Substrate Topology Information Availability and its Accuracy on Provider-independent Overlay Performance

In this thesis, we have summarized the methodologies that can be employed to perform the substrate topology discovery function for the ROMCA overlay. Based on this, we have investigated the impact of the availability and accuracy of such information on ROMCA overlay performance. To be more specific, we have examined the impact of IP aliases and the presence of anonymous routers in the tracerouted paths for application mapping. The following main conclusions can be drawn:

- Substrate topology discovery for the ROMCA overlay nodes can be implemented using traceroute. However, there are lots of issues that need to be dealt with and the inferred topology is usually incomplete/inaccurate.

- Through extensive examination of both real-network and synthetic topologies, we have demonstrated that substrate topology information is of crucial importance when attempting to obtain good performance with the proposed heuristic for application mapping with both resilience and QoS guarantees;

- For the AR resolution techniques we have compared, the Neighboring Matching resolution algorithm is the best in terms of reducing the overlap between the working and backup paths in the application mapping solution.

8.3 Future Work

Based on the work that have been carried out in the thesis, there are several topics that could be considered for future work:

- **Overlay Topology Construction**

1. We have discussed the ability of various overlay topology construction algorithms for providing resilience against substrate failures. This is based on the assumption that all the overlay nodes can obtain up-to-date information at the time of decision-making. However, this depends on the setting determining the monitoring and updating interval in the overlay layer. An investigation of different monitoring settings might be worthwhile to examine the trade-off between responsiveness and overhead.

2. The overlay node set is fixed for our evaluations although we do consider several alternatives. It would be interesting to investigate how the selection of overlay node locations affects the overlay performance. Since there are vast numbers of nodes that can meet the substrate degree requirement and can be considered to be overlay nodes, research focusing on network simplification done as in [146] might be helpful in reducing the size of this problem.

- **Application Mapping with Enhanced Resilience and QoS**

1. We have proposed a heuristic for solving the application mapping with resilience and QoS guarantees. It is fast when there is a solution, However, when no solution exists, it takes a long time until all possibilities are exhausted. A straightforward solution to this issue is to assign a time limit and stop searching when this limit is reached. However, other techniques, such as Monte Carlo methods, have been reported to perform fairly quick in solving decision-making problems [147]. Therefore, it might be helpful to make the heuristic more computationally efficient in the searching process.

2. According to the analysis of the ILP solution as compared to the heuristics, the ILP solution can obtain much lower delay performance. Therefore, one interesting topic that can be pursued is to investigate the possibility of improving the proposed heuristic in terms of providing better QoS whilst ensuring non-overlap working and backup paths.

- **The Impact of Substrate Network Availability and Accuracy on Provider-independent Overlay Performance**

We have investigated the impact of the IP aliases and the presence of ARs in the tracerouted paths on the overlay performance. There are a couple of topics that can be further pursued in regard to:

1. Examining the impact of using inference methods with less overhead as summarized in Chapter 6 on the performance of overlay in providing application mapping. Obviously, this will impact on the accuracy of the inferred substrate topology and may have different results than those obtained in this thesis given pairwise tracerouting information.

2. Examining the usefulness of tomography-based methods for obtaining more accurate substrate topologies. Tomography-based methods cannot be exploited directly for obtaining complete substrate network information due to their high computational complexity. However, if partial information obtained using tomography methods is incorporated, this might help improve the accuracy of the inferred substrate topology when the traceroute tool is not usable (i.e. the first-hop router discards all the tracerouting messages). This extension may further improve the performance of application mapping with both resilience and QoS guarantees.

Appendix A

Discussion of the LO-MARG Algorithm

A.1 Overview

For the proposed LO-MARG algorithm, Simulated Annealing (SA) is employed to find a near-optimal solution. The following issues are considered and discussed: (1) how to select the simulated annealing parameters so as to ensure a better solution (2) what is the best cost function.

A.2 SA Procedure and Parameter Selection

As for the first problem, two simulated annealing procedures categorized according to stopping criterion given in [148] are discussed. These two procedures are illustrated in Figure A.1 [1] and the notations used in this appendix are provided in Table A.1.

According to the discussion of parameters settings for the simulated annealing heuris-

---

[1] The simulated annealing procedure SA2 is explained in [149]. the Boltzman distribution mentioned in this figure is explained in Equation (A.2).
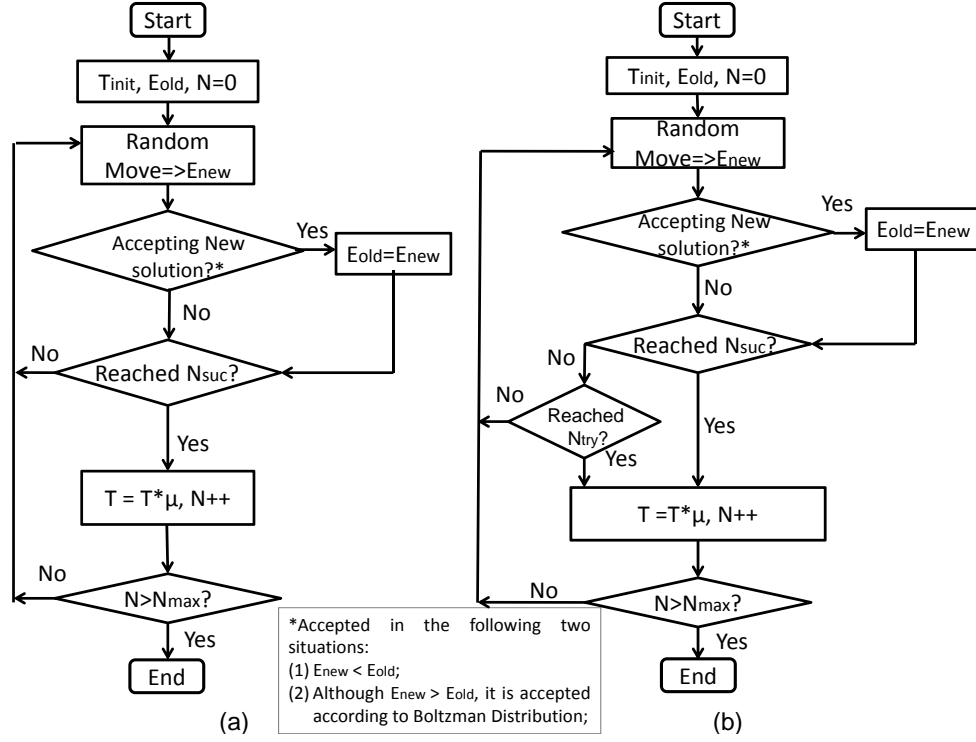
Figure A.1: Simulated annealing procedures: (a) procedure I (SA1); (b) procedure II (SA2)

Table A.1: Notation and definition

Notation	Definition
T	Current Temperature
$T_{initial}$	Initial Temperature
S	Current iteration Number
E_{old}	The energy value of last iteration
E_{new}	The energy value of current iteration
N_{try}	Maximum iteration number of trials in each round (temperature)
N_{suc}	Maximum number of successful trials in each round
N_{max}	Maximum number of rounds
N	Current iteration number
N_{OG}	The number of overlay node
R	A random value with the range $(0, 1)$
μ	Cooling Rate

tic given in the survey [148], all the settings (including the value of initial temperature, cooling schedule, number of iterations to be performed at each temperature and stopping criteria to terminate the algorithm) depend on the nature of the problem. However, the

following general guidelines are considered here:

- Initial temperature $T_{initial}$ should be considerably larger than the largest ΔE encountered, where $\Delta E = E_{new} - E_{old}$. We follow one of the adaptive methods summarized in [148] to determine the initial temperature. To be more specific,

$$T_{initial} = \frac{\Delta E_{init}^{max}}{ln(\chi_{init})}, \, where \, i \in (1, N_{init}) \tag{A.1}$$

  Where ΔE_{init}^{max}, χ_{init} and N_{init} are the maximum range of change recorded, the acceptance ratio obtained during the initial temperature decision process and the number of iteration employed for this process, respectively.

- Cooling rate μ is set between 0.8 and 0.99.

For a given simulation (i.e. 30 overlay nodes, overlay node degree of 5 with the Skitter-based topology and cost function FF2), 10 distinct parameter selections for each of these procedures are tested. The configuration of the SA parameters for each setting is explained in detail in Table A.2. In both procedures, the acceptance probability Pr of a new solution can be calculated using the following expression:

$$Pr(E_{new}) = \begin{cases} 1 & \text{if } E_{new} < E_{old} \\ 1 & \text{if } exp[-(E_{new} - E_{old})/T] > R, \text{ where } T = T_{init} \times \mu^S \\ 0 & \text{otherwise} \end{cases} \tag{A.2}$$

The metrics used to evaluate the SA procedures are listed as follows:

- Total iteration number (S_{total}). Theoretically, it should be calculated as follows[2]:

$$S_{total} = \begin{cases} N_{max} \times N_{try} & \text{For SA1} \\ < N_{max} \times N_{try} & \text{For SA2} \end{cases} \tag{A.3}$$

---

[2]For the SA2 procedure, only the upper bound can be obtained.

Table A.2: Parameter configuration for SA procedure comparison

No.	Proc.	$T_{initial}$	N_{max}	μ	N_{try}	N_{suc}
1	SA1	T_{init}	100	0.85	40^*N_{OG}	—
2	SA1	T_{init}	100	0.85	70^*N_{OG}	—
3	SA1	T_{init}	100	0.9	70^*N_{OG}	—
4	SA1	T_{init}	100	0.92	70^*N_{OG}	—
5	SA1	T_{init}	150	0.92	70^*N_{OG}	—
6	SA1	T_{init}	200	0.95	70^*N_{OG}	—
7	SA1	2^*T_{init}	100	0.92	70^*N_{OG}	—
8	SA1	T_{init}	100	0.9	40^*N_{OG}	—
9	SA1	T_{init}	100	0.93	70^*N_{OG}	—
10	SA1	T_{init}	100	0.91	70^*N_{OG}	—
11	SA2	T_{init}	100	0.85	40^*N_{OG}	20^*N_{OG}
12	SA2	T_{init}	100	0.85	70^*N_{OG}	20^*N_{OG}
13	SA2	T_{init}	100	0.85	70^*N_{OG}	30^*N_{OG}
14	SA2	T_{init}	100	0.9	70^*N_{OG}	30^*N_{OG}
15	SA2	T_{init}	150	0.9	70^*N_{OG}	30^*N_{OG}
16	SA2	T_{init}	100	0.92	70^*N_{OG}	30^*N_{OG}
17	SA2	2^*T_{init}	150	0.92	70^*N_{OG}	30^*N_{OG}
18	SA2	T_{init}	200	0.95	70^*N_{OG}	30^*N_{OG}
19	SA2	T_{init}	200	0.92	100^*N_{OG}	30^*N_{OG}
20	SA2	T_{init}	150	0.92	70^*N_{OG}	40^*N_{OG}

- the best overlapping sum OL_{sum} value and the iteration number S_{best} when this value is first obtained;

- the relative resilience performance of the overlay topology obtained using the LO-MARG algorithm;

The results are illustrated in Figure A.2 and Figure A.3 [3]. According to the simulation results shown in Figure A.2, Configuration No. 10 works the best for the SA1 procedure whilst Configuration No. 16 is best for the SA2 procedure. Generally, a higher iteration number and initial temperature will result in higher computational complexity. Conversely, the computation complexity can be decreased by choosing a lower number of maximum rounds for the annealing process and/or iteration numbers in each round but this results in higher risk of overlap due to insufficient searching of the solution space.

---

[3]Only the resilience performance of three typical configurations is presented.
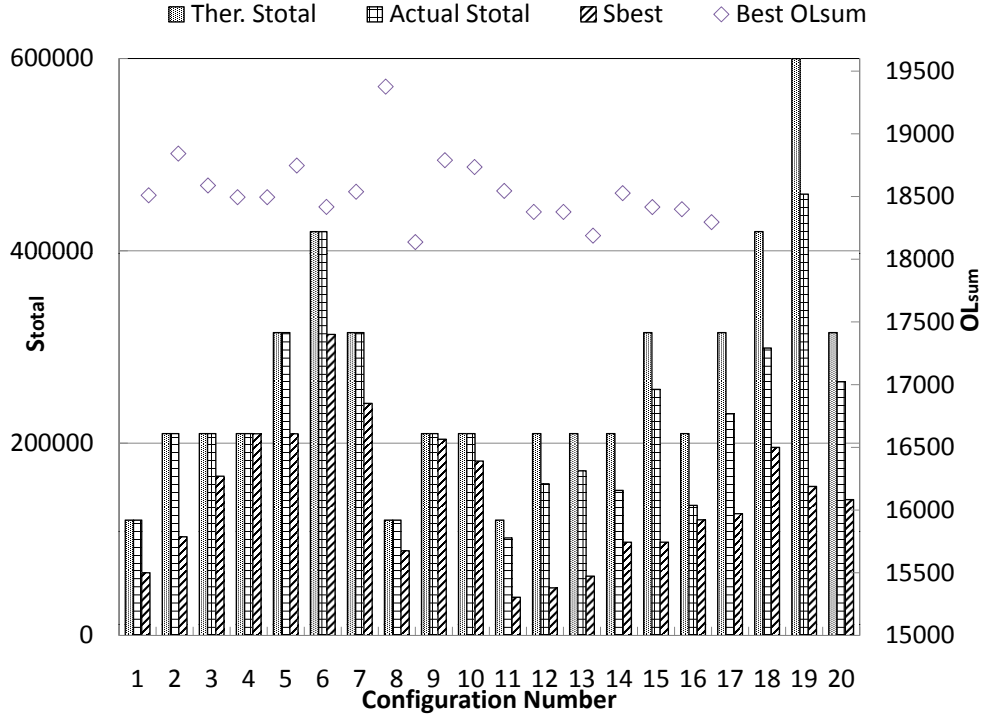
Figure A.2: Simulation results of different parameter configurations for two SA procedures

For both procedures, an appropriate setting of cooling rate as well as the iteration number is needed in order to obtain a better solution. According to Figure A.2, the SA1 procedure generally incurs much higher computational effort than that of SA2 when similar overlapping sum metrics are obtained. For instance, although the lowest overlapping sum is achieved by Configuration No. 10 with the SA1 procedure, its iteration times are 55% higher than that of Configuration No. 16 with the SA2 procedure. Moreover, as shown in Figure A.3, a comparatively high relative resilience performance can be achieved using Configuration No. 16 to that of Configuration No. 10. Thus, unless otherwise stated, Configuration No. 16 is employed with SA2 for the simulations with the same settings as discussed here. However, we are aware that configuration changes may result in worse performance and thus change the SA configuration when necessary as explained in Chapter 4 and Appendix B.
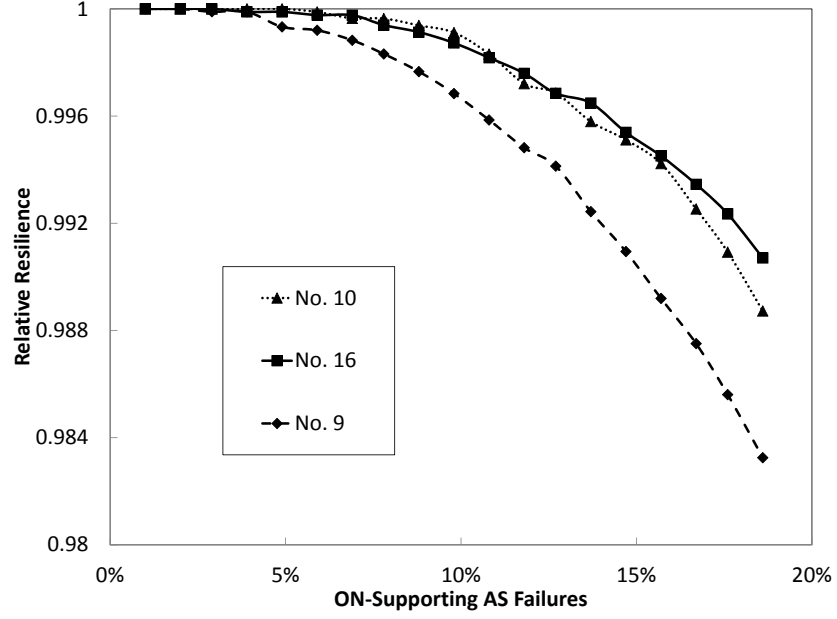
Figure A.3: Relative resilience performance of three parameter configurations

A.3   Cost Function Comparison

As for the cost function employed with the LO-MARG algorithm, four different alternatives are considered as listed below:

$$FF_1 = \frac{OL(L_I, L_J)}{\mid L_I \mid} \; where \mid L_I \mid \leq \mid L_J \mid \tag{A.4}$$

$$FF_2 = \mid L_I \mid \times \mid L_J \mid \times OL(L_I, L_J) \tag{A.5}$$

$$FF_3 = OL(L_I, L_J) \tag{A.6}$$

$$FF_4 = \begin{cases} 1 & \text{if } OL(L_I, L_J) \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{A.7}$$

Remarks:

- Cost function FF1 aims to count the percentage of overlap between a pair of virtual links;

- Cost function FF2 counts the weighted number of substrate nodes overlapping between a pair of virtual links. The weight counts the number of substrate hops covered by both virtual links;

- Cost function FF3 only counts the number of substrate nodes in common between both virtual links;

- Cost function FF4 is a binary variable that denotes whether two virtual links overlap or not.

- Cost function FF2 and FF3 are more accurate in terms of capturing the overlapping characteristics whilst the other two cost functions have filtered information. The advantage of cost function F4 is that it does not need complete underlying topological information.

We have carried out simulations with Skitter-based and Whois-based topologies using 30 and 50 overlay nodes. The results with 50 overlay nodes and overlay node degree of 5 are shown in Figure A.4. According to the simulation results as depicted in this figure, it can be seen that the best fitness function is FF2 and the worst is FF4. However, the

(a) Skitter-based   (b) Whois-based
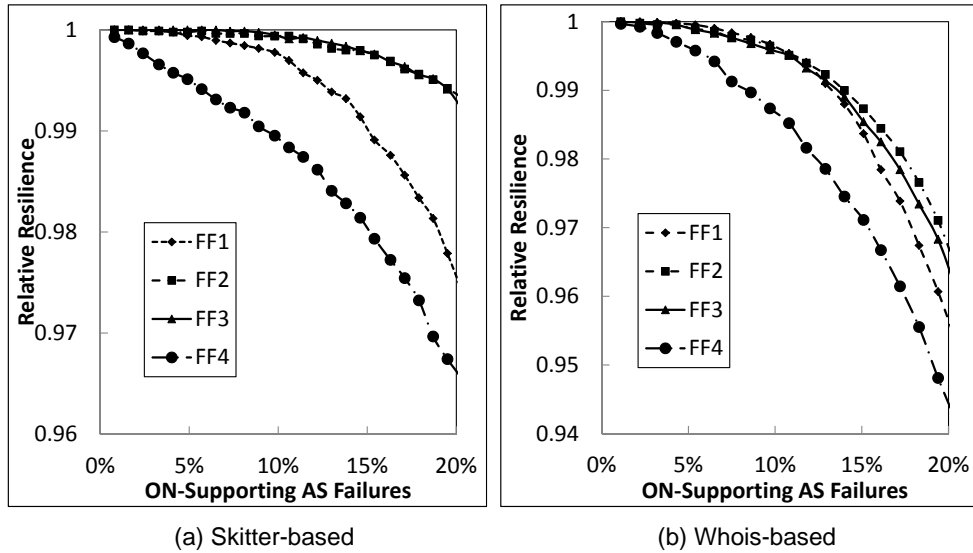
Figure A.4: Cost function comparison

figures show that different fitness functions yield similar performance when there is a low number of failures. Unless otherwise stated, FF2 is employed as the cost function for the SA-based LO-MARG algorithm.

A.4 Summary

In this chapter, we have presented the analysis of the SA procedure and parameter selection process for the LO-MARG algorithm. Moreover, different cost functions are discussed. Configuration No. 16 together with cost function FF2 is chosen for the LO-MARG algorithm use as a result.

Appendix B

Simulation Platforms and Verification

B.1   Overview

The simulation platforms employed in Chapter 4, Chapter 5 and Chapter 7 are built by the author. This appendix first explains the design of these platforms. Then, code verification of these platforms and justification of some typical simulation settings and assumptions are presented.

B.2   Overall Design of the Simulation Platforms (SP)

B.2.1   SP1: Overlay Topology Construction for Providing Resilience Service

The main modules of this simulator include:

- **Substrate network importing module**: It reads topology information using specific formatting and calculates pairwise end-to-end paths if not given. Acceptable formatting includes a link list file and topology files with delay, link and AS information generated by the GT-ITM topology generator.

- **Overlay node selection module**: It selects a specified number of eligible substrate nodes for use in overlay topology construction. It can randomly choose a certain number of substrate nodes or follow certain strategies during the selection process.

- **Topology construction module**: It implements various overlay topology construction algorithms, including the proposed ones (i.e. LO-MARG, EDL-KMST and RM-RG) and the existing ones (i.e. FM, TKMST, TKRC and KRC) given accurate or inferred substrate topology information.

- **Performance evaluation module**: It evaluates the performance of various overlay construction topologies under different failure models and collects statistics.

The flow chart of this simulator (SP1) is illustrated in Figure B.1 (a).

B.2.2 SP2: Application Mapping with Resilience and QoS Guarantees

The main modules of this simulator include:

- **Substrate network importing module**: It reads topology information using specific formatting and calculates pairwise end-to-end paths if not given. Acceptable formatting includes topology files with delay, link and AS information generated by the GT-ITM topology generator and real-network topology information with both pairwise delay and pairwise overlap information.

- **Overlay node selection module**: It randomly selects a specified number of eligible substrate nodes for use in application mapping or uses the end hosts given

in the real-network dataset.

- **Application mapping module**: It implements the proposed heuristic (i.e. pQoSMap) and the best existing one (i.e. QoSMap) given exact or inferred topologies.

- **Performance evaluation module**: It evaluates the performance of the mapped overlay solution and collects statistics including D_{avg}, O_{wb} and C_r and so forth.

The flow chart of this simulator (SP2) is illustrated in Figure B.1 (b).

B.2.3 SP3: Substrate Topology Inference

The main modules of this platform include:

- **Network importing module**: It reads topology information using specific formatting and calculates pairwise end-to-end paths if not given. Acceptable formatting includes the topology file with delay, link and AS information the GT-ITM topology generator or the real-network traceroute dataset provided by the iPlane Project.

- **Initial Processing module**: This module is only for use with the real network dataset. It selects a subset of nodes that have tracerouting information given the traceroute dataset. The criteria used during the subset selection process includes (1) the chosen nodes should have complete traceroute paths (with or without AR anonymous nodes) to all the other chosen nodes; (2) it should not include any entry with no IP addresses.

- **Anonymous node generation module**: It assigns certain number of nodes to be anonymous nodes of T-1 and T-3 as defined in Chapter 6. This is only used for synthetic topologies.

- **IP Alias resolution module**: It resolves IP aliases using the analytic method presented in [116].
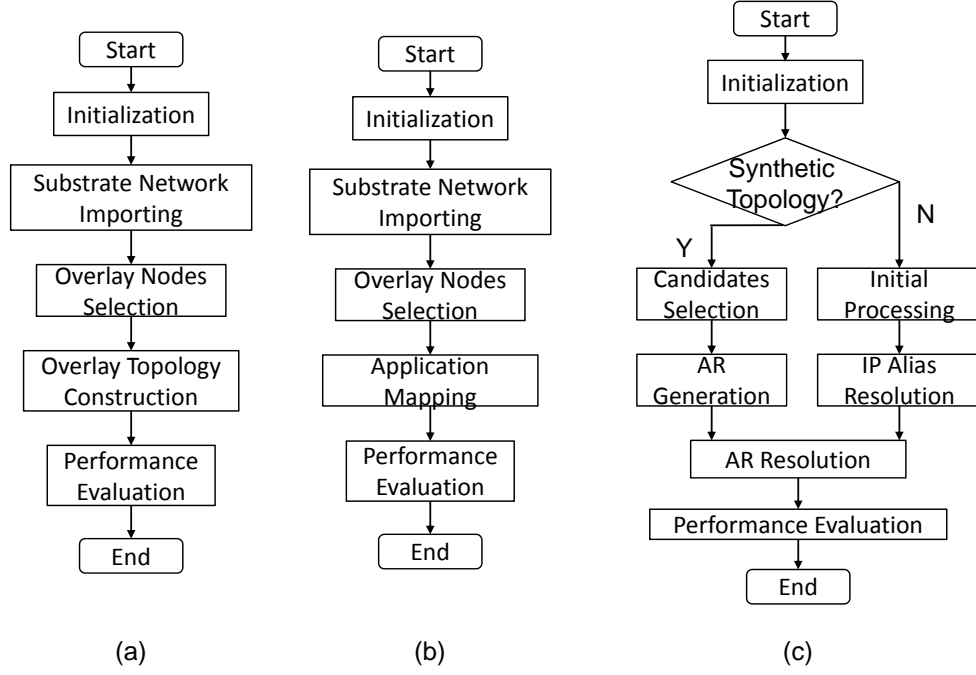
Figure B.1: Overall design of simulation platforms: (a) SP1: overlay topology construction; (b) SP2: application mapping; (c) SP3: substrate topology inference

- **AR resolution module**: It resolves AR nodes using different AR resolution methods presented in [104].

- **Performance evaluation module**: It evaluates the performance of various techniques with inferring topologies and collects evaluation metrics such as the number of nodes, the number of links, AR ratio as well as information concerning the inferred substrate topology for overlay use such as E2E delay and path overlap.

The flow chart of this simulator (SP3) is illustrated in Figure B.1 (c).

B.2.4   Relationship Among the SPs

As described in Section B.2.1 to B.2.3, the three platforms either share similar modules or the outcome of one platform can be used as the input to another. The relationship among these three platforms is depicted in Figure B.2.
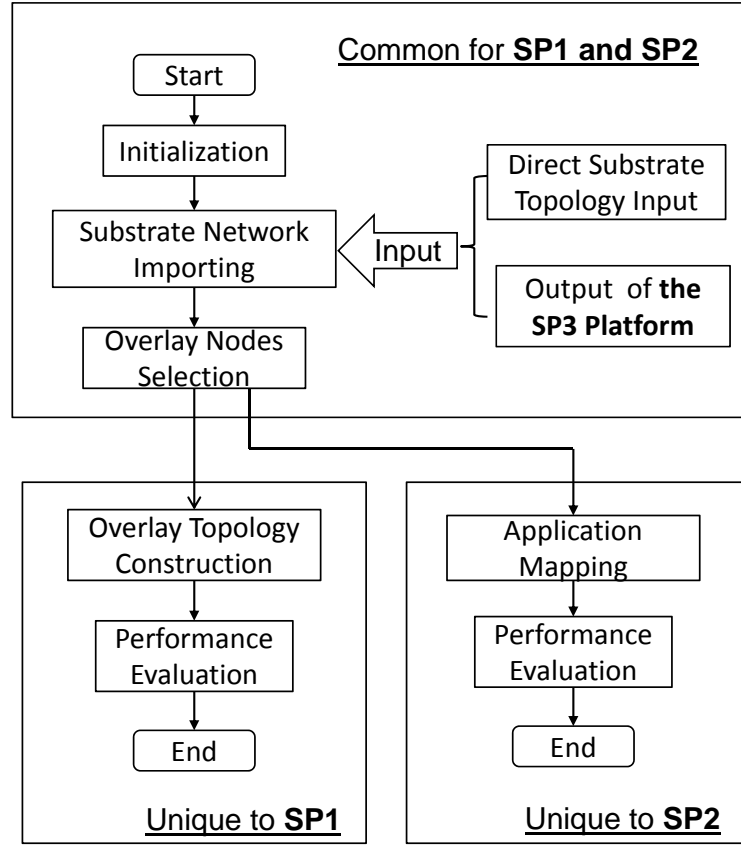
Figure B.2: The relationship among all the SPs

B.2.5 Design of the CPLEX-based ILP Solution

The enhanced ILP model proposed in Chapter 5 is solved using IBM ILP solver CPLEX. We adopt the method utilized in [82] to simplify the complexity of solving the ILP model. To be more specific, each application node establishes the connections to the candidate overlay nodes so as to form a single topology in the ILP solution. Matlab is exploited to generate the input format required by this ILP solver using the objective functions and constraints specified in Chapter 5.

All the variables are defined in Chapter 5 and explained in their linear form except $B_{m'}$. We thus explain the implementation of this variable here in the ILP model for completeness. Three new variables $B_{m'}^{AB}$, $BW_{m'}$ and $M_{m'}$ are introduced.

$B_{m'}^{AB}$ is equal to 1 if an overlay node $\nu_{m'}$ is used as an intermediate overlay node in the backup path for L_{AB}, otherwise 0. This new variable can be expressed using the following equation:

$$
B_{m'}^{AB} = \begin{cases} 1 & \text{if } \exists L_{AB}, \sum_{\{\nu_{j'}\}} B_{j'm'}^{AB} = 1 \cap \sum_{\{\nu_{k'}\}} B_{m'k'}^{AB} = 1, \text{ (Intermediate)} \\ 0 & \text{if } \exists L_{AB}, \sum_{\{\nu_{j'}\}} B_{j'm'}^{AB} = 1 \cap \sum_{\{\nu_{k'}\}} B_{m'k'}^{AB} = 0, \text{ (Destination)} \\ 0 & \text{if } \exists L_{AB}, \sum_{\{\nu_{j'}\}} B_{j'm'}^{AB} = 0 \cap \sum_{\{\nu_{k'}\}} B_{m'k'}^{AB} = 1, \text{ (Source)} \\ 0 & \text{otherwise} \end{cases} , \forall \nu_{m'}, \forall L_{AB}
$$

(B.1)

which can be linearized using the same method for $\Phi_{m'n,p'q'}$ linearization presented in Chapter 5.

The second variable $BW_{m'}$ is a binary and is equal to 1 when it is used as a backup (note that it can be used as the hosting node as well); otherwise, 0. Therefore, $BW_{m'}$ can be expressed as:

$$
B_{m'} = \begin{cases} 1 & \text{if } \exists B_{m'}^{AB} = 1 \ \forall L_{AB} \\ 0 & \text{otherwise} \end{cases} , \forall \nu_{m'}
$$

(B.2)

which can be linearized as follows:

$$
BW_{m'} \leq \sum_{L_{AB}} B_{m'}^{AB}, \quad \forall \nu_{m'}
$$

(B.3)

$$
BW_{m'} \geq B_{m'}^{AB} \quad \forall \nu_{m'}, L_{AB}
$$

(B.4)

The third variable $M_{m'}$ is a binary and is equal to one if $\nu_{m'}$ is used as a hosting node. Its formulation is similar to that of the $BW_{m'}$ and thus omitted here.

Therefore, $B_{m'}$ can be expressed as:

$$
B_{m'} = \begin{cases} 1 & \text{if } BW_{m'} = 1 \text{ and } M_{m'} = 0 \\ 0 & \text{otherwise} \end{cases} , \forall \nu_{m'}
$$

(B.5)

The relationship above can be expressed linearly using the following inequalities:

$$B_{m'} \leq 1 - M_{m'} \quad \forall \nu_{m'} \tag{B.6}$$

$$B_{m'} \geq BW_{m'} - M_{m'}, \quad \forall \nu_{m'} \tag{B.7}$$

$$B_{m'} \leq BW_{m'}, \quad \forall \nu_{m'} \tag{B.8}$$

The correctness of the ILP solution is verified and explained in section B.3.

B.3 Code Verification

As the credibility of these platforms is important groundwork for this thesis, the simulation code has been debugged module by module with the help of breakpoints and embedded error reporting code. Moreover, they are also verified exploiting one or more of the following methods where appropriate:

- Testing a module/platform given a small input and comparing its output with the hand-calculated output;

- Comparing the result of one algorithm with the optimal solution obtained through other means given the same simulation setting (For example, the optimality of the SA-based LO-MARG algorithm compared with that of the optimal solution obtained by the "brute-force" method);

- Comparing the result of one simulator with that of the other (For example, comparing the heuristic with the ILP optimal solution for application mapping.);

- Comparing the result of a module/platform with results published by the work of other researchers;

(a) Small Network (b) Setting and Results
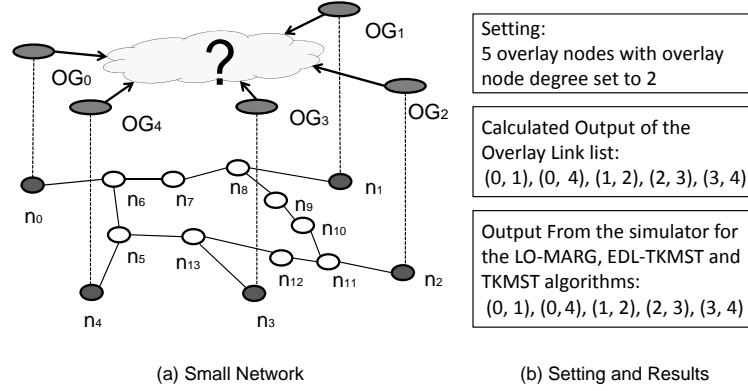
Figure B.3: Verification of overlay topology algorithms in a small network

In the following, we present the verification of some key modules of these simulation platforms exploiting the above-mentioned methods.

B.3.1 Verification of Overlay Construction Algorithms

Besides line by line debugging, we also use a small-network scenario to verify the correctness of the algorithms with a deterministic output [1]. The small network together with the output is shown in Figure B.3 and the result shows that the output of the simulator is the same as the result calculated by hand.

As for the optimality of the SA-based LO-MARG algorithm, we have tested it with an overlay configuration of a small node population where the "brute-force" method can be applied. One setting[2] of the verification and the simulation result is explained in Table B.1. As shown in the result, the LO-MARG algorithm can obtain the optimal solution with much lower complexity as compared to that of the "brute-force" method with the configured simulation setting.

---

[1] Algorithms decided with the help of random generated numbers cannot be evaluated with this method since the output depends on the random number generated.

[2] We have also tried with the number of overlay nodes set to 6 and the result is similar and is thus not presented here.

Table B.1: Parameter setting for verifying the LO-MARG algorithm

Parameter Setting	
Substrate Topology	Skitter-based
Overlay Node Number	10
Overlay Node Degree	5
Cost Function	FF2
SA Setting	Configuration No. 16
Results	
Brute-Force	Iteration=3628800, Lowest $OL_{sum} = 2130$
LO-MARG	Iteration = 27618, Lowest $OL_{sum} = 2130$

B.3.2 Verification of the ILP and the Heuristic for Application Mapping

For the correctness of the enhanced ILP model, we have exploited a small network with different network settings for verification. Moreover, the correctness of the heuristic is also checked by comparing its output with that of the ILP solution for the same-network scenarios since the ILP solution can provide the optimal solution.

The small network is depicted in Figure B.4 and the results with three different settings are presented in Figure B.5, Figure B.6 and Figure B.7 respectively. With the scenario explained in Figure B.5, the ILP and the two heuristics can all obtain the optimal solution. However, in a different scenario, both heuristics do not achieve the lowest delay value whilst the ILP solution can do so at the expense with an extra overlay node involved in the mapping solution. We have also changed the weight factors of the ILP objective function and the result is presented in Figure B.8. The ILP solution can also achieve no additional cost measured in the number of overlay nodes used for backup only if required. In the third case, the *QoSMap* heuristic finds a solution with 100% overlap between the working and backup paths whilst the proposed heuristic and the ILP solution can avoid overlap with a a slightly higher value of average delay. In conclusion, we verified the correctness of the implementation of both the ILP solution and the two heuristics at least for these scenarios.
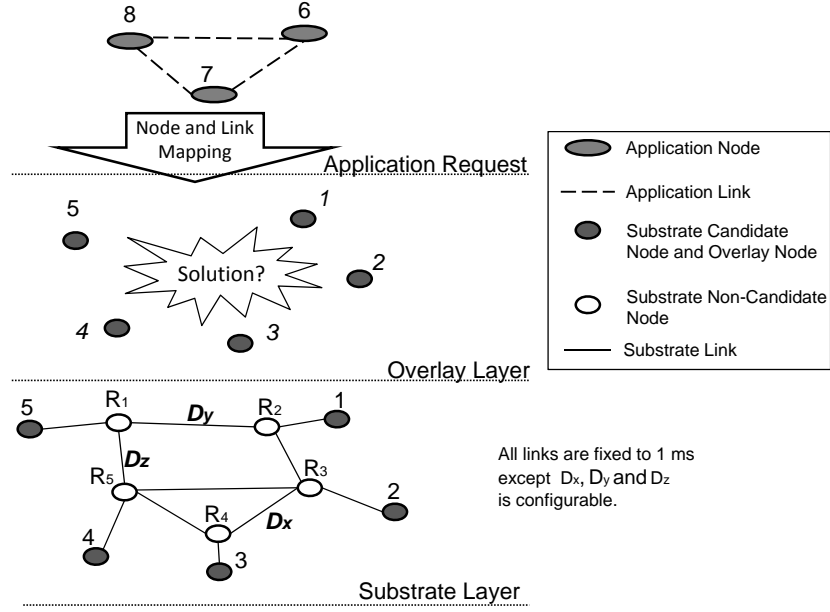
Figure B.4: Verification scenarios for ILP and heuristics

Setting:	Dmax= 6ms, Dx=Dy=Dz=1ms (For ILP, alpha=0.98, beta= gamma=0.01)	
QoSMap	**pQoSMap**	**Enhanced ILP Model**
=Dall=9.00	=Dall=9.00	\<variable name="**Dl**" index="0" value="**9**"/>
=Dave=3.00	=Dave=3.00	\<variable name="**Olww**" index="1" value="**0**"/>
		\<variable name="**BNWnum**" index="2" value="**0**"/>
OL[1]=0	OL[1]=0	
OL[2]=0	OL[2]=0	\<variable name="M_6_1" index="232" value="0"/>
OL[3]=0	OL[3]=0	\<variable name="M_7_1" index="233" value="0"/>
=OLsum=0	=OLsum=0	\<variable name="M_8_1" index="234" value="0"/>
=OL%=0.00	=OL%=0.00	\<variable name="M_6_2" index="235" value="0"/>
		\<variable name="M_7_2" index="236" value="0"/>
mapped[6]=2	mapped[6]=2	**\<variable name="M_8_2" index="237" value="1"/>**
mapped[7]=4	mapped[7]=4	**\<variable name="M_6_3" index="238" value="1"/>**
mapped[8]=3	mapped[8]=3	\<variable name="M_7_3" index="239" value="0"/>
		\<variable name="M_8_3" index="240" value="0"/>
=C_r=0	=C_r=0	\<variable name="M_6_4" index="241" value="0"/>
======	======	**\<variable name="M_7_4" index="242" value="1"/>**
		\<variable name="M_8_4" index="243" value="0"/>
		\<variable name="M_6_5" index="244" value="0"/>
		\<variable name="M_7_5" index="245" value="0"/>
		\<variable name="M_8_5" index="246" value="0"/>

Figure B.5: Results for scenario 1

B.3.3 Verification of Substrate Topology Inference

The simulation platform for substrate topology inference consists of two main modules, namely, IP alias resolution and anonymous router resolution. Hence, we verify their

Setting: Dmax= 10ms, Dx= 8ms, Dy=1ms, Dz=2ms (for ILP, alpha=0.98, beta= gamma=0.01)		
QoSMap	**pQoSMap**	**Enhanced ILP Model**
=Dall=11.00	=Dall=11.00	\<variable name="**Dl**" index="0" value="**10**"/\>
=Dave=3.67	=Dave=3.67	\<variable name="**Olww**" index="1" value="**2**"/\>
		\<variable name="**BNWnum**" index="2" value="**1**"/\>
OL[1]=0	OL[1]=0	
OL[2]=0	OL[2]=0	\<variable name="M_6_1" index="260" value="0"/\>
OL[3]=0	OL[3]=0	\<variable name="M_7_1" index="261" value="0"/\>
=OLsum=0	=OLsum=0	\<variable name="M_8_1" index="262" value="0"/\>
=OL=0.00	=OL=0.00	**\<variable name="M_6_2" index="263" value="1"/\>**
		\<variable name="M_7_2" index="264" value="0"/\>
mapped[6]=1	mapped[6]=1	\<variable name="M_8_2" index="265" value="0"/\>
mapped[7]=5	mapped[7]=5	\<variable name="M_6_3" index="266" value="0"/\>
mapped[8]=4	mapped[8]=4	**\<variable name="M_7_3" index="267" value="1"/\>**
		\<variable name="M_8_3" index="268" value="0"/\>
=C_r=0	=C_r=0	\<variable name="M_6_4" index="269" value="0"/\>
======	======	\<variable name="M_7_4" index="270" value="0"/\>
		\<variable name="M_8_4" index="271" value="1"/\>
		\<variable name="M_6_5" index="272" value="0"/\>
		\<variable name="M_7_5" index="273" value="0"/\>
		\<variable name="M_8_5" index="274" value="0"/\>

Figure B.6: Results for scenario 2

Setting: Dmax= 10ms, Dx= Dy=3ms, Dz=1ms (for ILP, alpha=0.98, beta= gamma=0.01)		
QoSMap	**pQoSMap**	**Enhanced ILP Model**
=Dall=10.00	=Dall=12.00	\<variable name="**Dl**" index="0" value="**12**"/\>
=Dave=3.33	=Dave=4.00	\<variable name="**Olww**" index="1" value="**0**"/\>
		\<variable name="**BNWnum**" index="2" value="**0**"/\>
OL[1]=2	OL[1]=0	
OL[2]=1	OL[2]=0	\<variable name="M_6_1" index="268" value="0"/\>
OL[3]=1	OL[3]=0	\<variable name="M_7_1" index="269" value="0"/\>
=OLsum=3	=OLsum=0	**\<variable name="M_8_1" index="270" value="1"/\>**
=OL%=1.00	=OL%=0.00	**\<variable name="M_6_2" index="271" value="1"/\>**
		\<variable name="M_7_2" index="272" value="0"/\>
mapped[6]=2	mapped[6]=2	\<variable name="M_8_2" index="273" value="0"/\>
mapped[7]=3	mapped[7]=5	\<variable name="M_6_3" index="274" value="0"/\>
mapped[8]=4	mapped[8]=1	\<variable name="M_7_3" index="275" value="0"/\>
		\<variable name="M_8_3" index="276" value="0"/\>
=C_r=0	=C_r=0	\<variable name="M_6_4" index="277" value="0"/\>
======	======	\<variable name="M_7_4" index="278" value="0"/\>
		\<variable name="M_8_4" index="279" value="0"/\>
		\<variable name="M_6_5" index="280" value="0"/\>
		\<variable name="M_7_5" index="281" value="1"/\>
		\<variable name="M_8_5" index="282" value="0"/\>

Figure B.7: Results for scenario 3

correctness individually.

- **IP Alias Resolution**

Only the real-network dataset needs to use the IP alias resolution function. In order to verify whether this function is implemented correctly, one example of the simulator's IP

Setting: Dmax= 10ms, Dx= 8ms, Dy=1ms, Dz=2ms (ILP Parameter Setting)	
alpha=0.98, beta= gamma=0.01	**alpha=gamma=0.01, beta=0.98**
<variable name="**Dl**" index="0" value="**10**"/>	<variable name="**Dl**" index="0" value="**11**"/>
<variable name="**Olww**" index="1" value="**2**"/>	<variable name="**Olww**" index="1" value="**0**"/>
<variable name="**BNWnum**" index="2" value="**1**"/>	<variable name="**BNWnum**" index="2" value="**0**"/>
<variable name="M_6_1" index="260" value="0"/>	**<variable name="M_6_1" index="260" value="1"/>**
<variable name="M_7_1" index="261" value="0"/>	<variable name="M_7_1" index="261" value="0"/>
<variable name="M_8_1" index="262" value="0"/>	<variable name="M_8_1" index="262" value="0"/>
<variable name="M_6_2" index="263" value="1"/>	<variable name="M_6_2" index="263" value="0"/>
<variable name="M_7_2" index="264" value="0"/>	<variable name="M_7_2" index="264" value="0"/>
<variable name="M_8_2" index="265" value="0"/>	<variable name="M_8_2" index="265" value="0"/>
<variable name="M_6_3" index="266" value="0"/>	<variable name="M_6_3" index="266" value="0"/>
<variable name="M_7_3" index="267" value="1"/>	<variable name="M_7_3" index="267" value="0"/>
<variable name="M_8_3" index="268" value="0"/>	<variable name="M_8_3" index="268" value="0"/>
<variable name="M_6_4" index="269" value="0"/>	<variable name="M_6_4" index="269" value="0"/>
<variable name="M_7_4" index="270" value="0"/>	<variable name="M_7_4" index="270" value="0"/>
<variable name="M_8_4" index="271" value="1"/>	**<variable name="M_8_4" index="271" value="1"/>**
<variable name="M_6_5" index="272" value="0"/>	<variable name="M_6_5" index="272" value="0"/>
<variable name="M_7_5" index="273" value="0"/>	**<variable name="M_7_5" index="273" value="1"/>**
<variable name="M_8_5" index="274" value="0"/>	<variable name="M_8_5" index="274" value="0"/>

Figure B.8: Comparison of weight factor settings on the ILP enhanced model
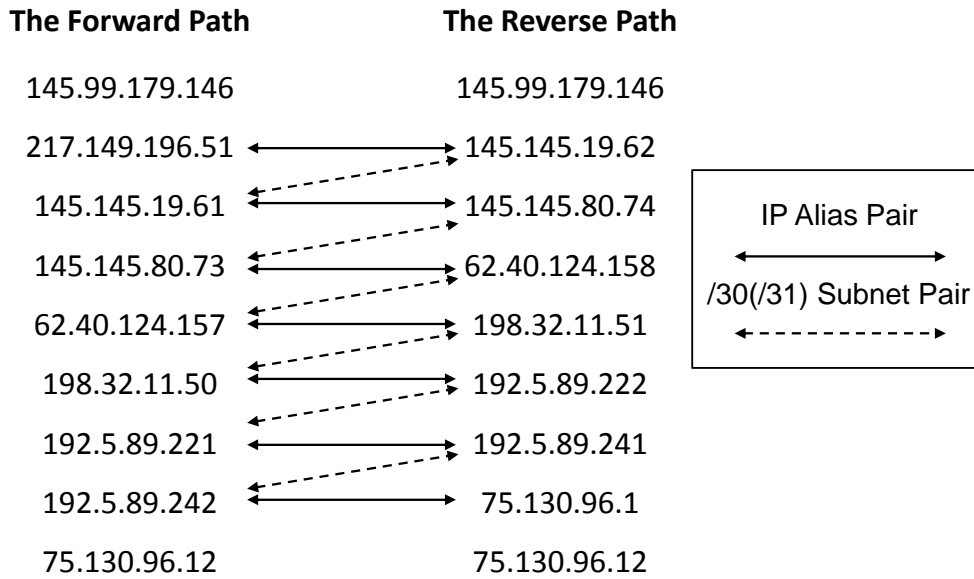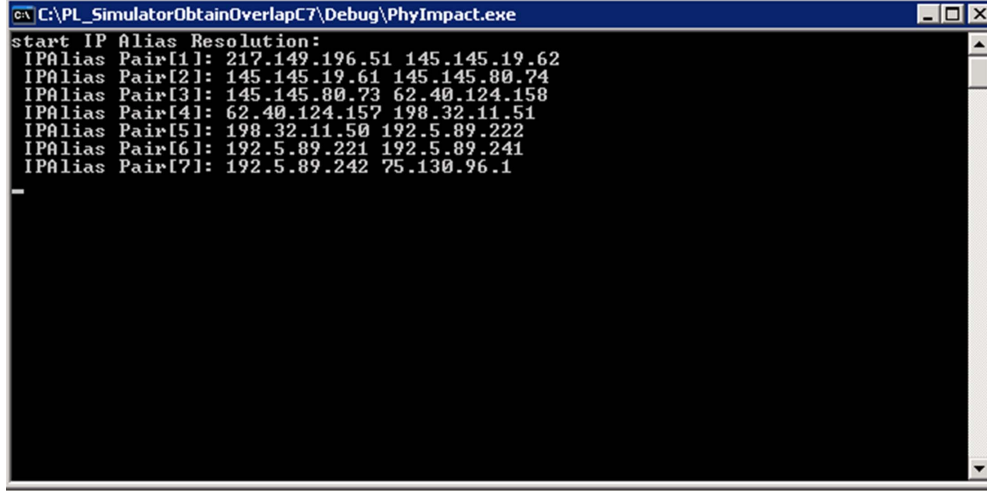
Figure B.9: Hand-calculated output of the IP alias pair identification

alias resolution function is printed in the output window. The simulation output of the IP alias resolution results for the path pair depicted in Figure B.9 is shown in Figure B.10. According to our own analysis of the IP alias pair for these two paths in Figure B.9, it matches the ones produced by the simulator. Therefore, the implementation of this function is considered to be correct.

Figure B.10: Corresponding simulation output of IP alias pair identification

- **AR Resolution**

In order to evaluate the correctness of different AR resolution techniques, we have compared the statistics obtained in our simulator with that of the work in [104] given both synthetic and real-network topologies.

Since we have no intention to evaluate the effectiveness of various techniques which have been provided by their work in obtaining a topology closer to the true one, here we use the following metrics:

1. **Node ratio**: The ratio of the number of nodes included between the inferred topology and in the true topology;

2. **Link ratio**: The ratio of the number of links included between the inferred topology and in the true topology;

3. **AR ratio**: The ratio of the number of AR nodes included between the inferred topology and the true topology;

4. **AR%**: The percentage of nodes that are ARs nodes in the inferred topology;

Table B.2: Evaluation of the inferred topologies exploiting different AR resolution methods

Metrics	Raw	IP	NM	GBI
For the synthetic topology (AR Ratio=10%)				
Node Ratio	16.94	2.14	1.2	1.25
Link Ratio	17.97	1.81	1.26	1.27
AR Ratio	177.24	13.62	3.27	3.76
AR%	94.62%	57.52%	24.5%	27.23%
For the real-network topology				
Node Number	11354	3347	3054	3078
Link Number	20435	6474	6181	6184
AR%	75.1%	15.51%	7.4%	8.1%

Since it is not possible to obtain the true topology information for the real-network dataset, we use the number of nodes, the number of links and AR% instead for the real network evaluation.

As shown in Table B.2, the data obtained for the synthetic topology are comparable to that provided in Table II and III of the work [104] and the work [44], respectively. The comparative values of the AR% metric for different AR resolution techinques in both the synthetic and real topologies are similar. Please note that the metric values in our network scenarios are different and generally larger because we infer the topology among a group of end nodes (including forward and reverse paths for a end node pair) whilst they only try to infer the topology from X sources to Y destinations where X is far smaller than Y. Furthermore, the AR% obtained in the synthetic topology is bigger than that in the real network because traceroute paths with all "*"s are avoided in the real-network dataset when we select a subset of 201 end-nodes. However, in the synthetic topologies, we still include this type of ARs since we have the true topology for complete evaluation.

B.4 Justification and Verification of Simulation Settings and Assumptions

In this section, we present justification and verification for some typical settings and assumptions made regarding simulation evaluation presented in this thesis.

B.4.1 Setting of the Random Failure Generation Iteration Number

The Mersenne Twister [150] random number generator has been implemented in the simulation platforms with functions to set different seeds and functions that implement different probability distributions. In Chapter 4, random failures are generated to evaluate the performance of various overlay construction algorithms. In this section, we examine whether setting the iteration number of random failures to 1000 as suggested by [48] is appropriate. In order to do so, we have changed the number of failure iterations to 2000 and compared the results. We present the results given the overlay node population and overlay node degree to 30 and 5, respectively, with the Whois-based AS-level topology and they are shown in Figure B.11 [3]. As shown in the results, we believe that the setting of 1000 random failures is reasonable since the relative error is within 0.6% using the results averaged over 2000 iterations as the baseline.

B.4.2 Assumption of Fixed Overlay Node Set in Overlay Construction for Providing Resilience Service

In Chapter 4, we assume that the overlay node set is fixed for performance evaluation and we have used re-scaled AS-level topologies as simulation input. Here, we have tried 30 different sets of overlay node given the same AS-level graphs. Depending on the overlay node set, the number of ASes included in the overlay is different, so we use the

---

[3]Since the graphs for different methods are similar, only the results for Full Mesh and LO-MARG algorithms are presented here for clarity.
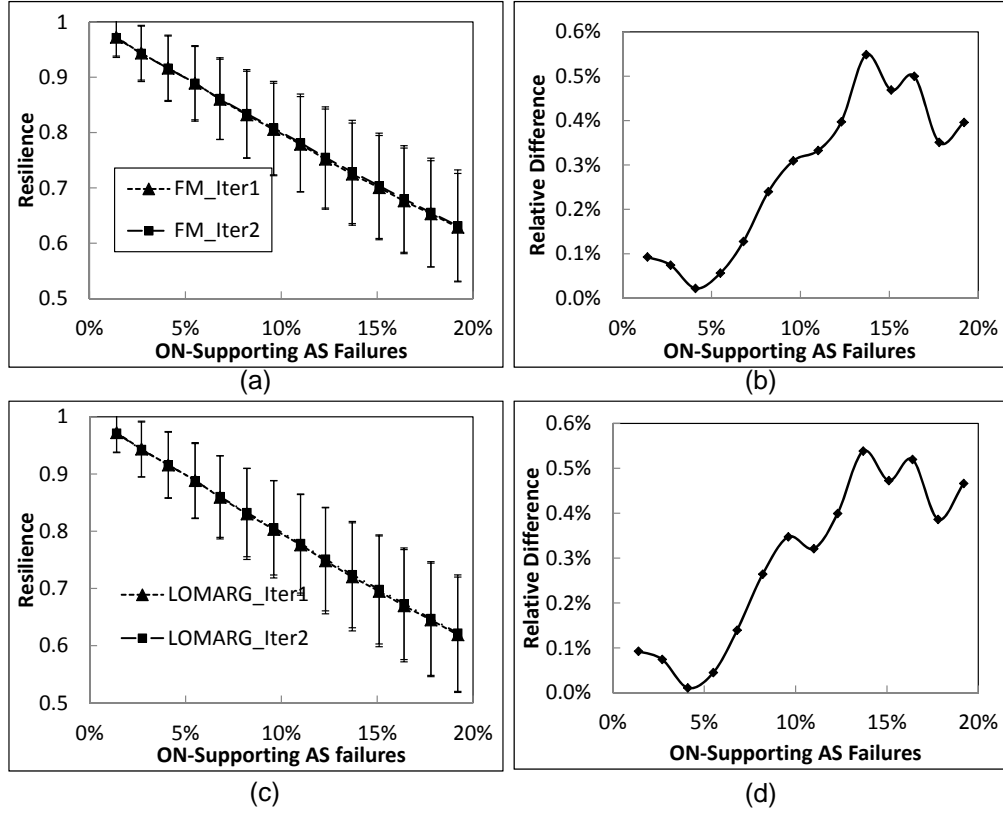
Figure B.11: Failure generation iteration number verification: (a) the resilience of the FM method over 1000 and 2000 (baseline) iterations; (b) the relative difference of the two results in (a); (c) the resilience of the LO-MARG method over 1000 and 2000 (baseline) iterations; (d) the relative difference of the two results in (c)

AS failure number as x-axis instead. The results are given in Figure B.12 [4]. As shown by the relative performance of different methods in the graphs, we believe that the results presented in Chapter 4 are typical. However, a discussion of the overlay node location is not discussed in this thesis and is considered future work.

_____

[4]The LO-MARG algorithm is not the best with the Skitter-based AS-level topology due to the setting of the simulated annealing parameter settings. Since the difference between the best one (EDL-KMST) and the LO-MARG algorithm is small, we choose not to go through the process presented in Appendix A again. However, it can be undertaken by following the parameter setting process described in Appendix A in order to achieve better performance with the LO-MARG algorithm.
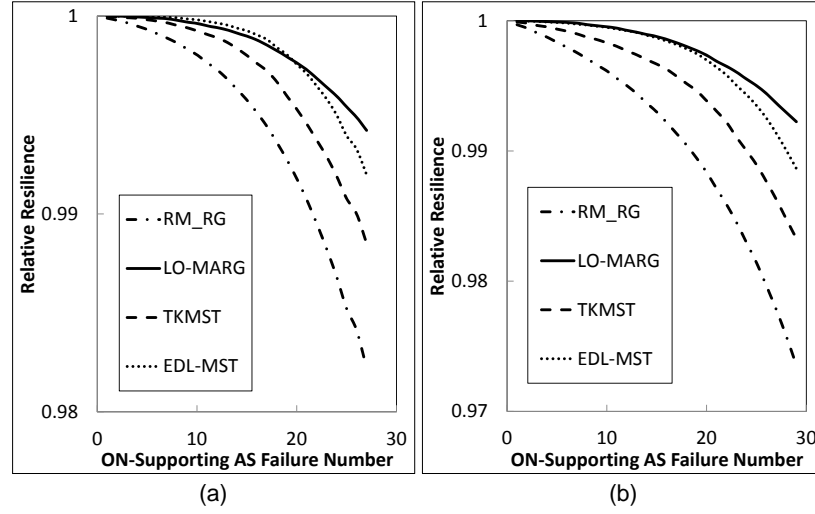
Figure B.12: Relative resilience of RM-RG, LO-MARG, TKMST and EDL-KMST averaged over 30 different OG node sets: (a) with the Skitter-based AS-level topology; (b) with the Whois-based AS-level topology

B.4.3 Impact of Regular Graphs with Different Girths in Overlay Construction for Providing Resilience Service

We investigate the impact of regular graphs with different girths here. One of the important properties of regular graphs is girth. Girth is defined as the length of the shortest polygon in the graph, where a polygon is a sequence of edges and vertices where no edge or vertex is included more than once except that the first vertex and last coincide [77]. Simulations are carried out using regular graphs with girth equals to 3 and 5 respectively, given 30 overlay nodes and overlay node degree of 5 using the Skitter-based AS-level topology. As shown in Figure B.13, there is no substantial difference (i.e. <0.1%) between the results with different girths when the AS Failure number is small (i.e. <10%). When the AS failure number increases, the regular graph with 5 can perform better as compared to the one with girth 3. Moreover, we have assumed that the regular graph is fixed as an input for simulation in the thesis. Hence, we can assume the results are consistent for all methods.

As for the RM-RG method, we fixed the iteration number to be 30 and we have also
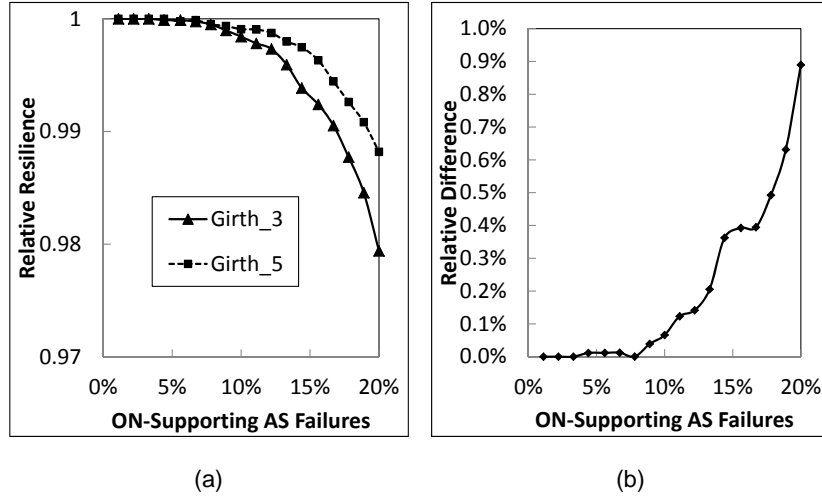
Figure B.13: The impact of regular graphs with different girths on the LO-MARG algorithm: (a) relative resilience; (b) relative difference using the results of regular graph with girth equal to 5 as the baseline

verified this against a setting of 100 and the difference in results is within 0.15%. Hence, we conclude a setting of 30 is reasonable.

B.4.4 Assumption of Fixed Overlay Node Set in Application Mapping with Resilience and QoS Guarantees

In Chapter 7, we assume the candidate overlay node set is fixed for the performance evaluation of the proposed heuristic with the synthetic topologies. In this section, we have examined changing the seed of the random number generator for the candidate overlay node selection process 30 times for one fixed application request scenario. Table B.3 provides the simulation settings and statistics. As provided in the table, the following observations can be made:

1. In general, the degree of overlap obtained with the existing heuristic is significant and the results presented in Chapter 7 are considered to be typical.

2. As for the delay performance, the proposed heuristic's results are slightly worse than that can achieved by the existing one. The reason that the proposed heuristic

Table B.3: Simulation settings and statistics for different overlay node sets for application mapping

Simulation Settings	
Application Node Size	30
Application Topo. Type	Full Mesh
Remaining Connectivity	100%
Overlay Node Size	80
Iteration Number	30
Simulation Results	
D_{avg}	
Methods	Value
$QoSMap$	(Average): 266.34, (Std. Deviation): 19.79
$pQoSMap$	(Average): 278.89, (Std. Deviation): 32.59
O_{wb}	
Methods	Value
$QoSMap$	(Average): 51.5% (Std. Deviation): 22.3%
$pQoSMap$	0
C_r	
Methods	Value
$QoSMap$	0
$pQoSMap$	≤ 2

has a wider delay performance than that of the existing one in terms of the delay performance is that it is affected by the topology features of the substrate during the mapping process whilst the existing heuristic does not consider substrate topology information. Again, we consider the results presented in Chapter 7 to be typical values for delay performance.

B.5 Summary

In this appendix, we have described the design of the simulation platforms used throughout this thesis in detail. Moreover, we have also presented verification of the self-built simulation platforms and some typical settings and assumptions.

References

[1] "Statistics about the Internet," http://www.internetworldstats.com/emarketing. htm.

[2] D. G. Andersen, "Improving end-to-end availability using overlay networks," Ph.D. dissertation, Massachusetts Institute of Technology, 2005.

[3] V. E. Paxson, "Measurements and analysis of end-to-end Internet dynamics," Tech. Rep., 1997.

[4] "The 16-bit AS number report," http://www.potaroo.net/tools/asn16/, retrieved on 22 Jul. 2010.

[5] N. Feamster, H. Balakrishnan, and J. Rexford, "Some foundational problems in interdomain routing," in *3rd ACM SIGCOMM Workshop on Hot Topics in Networks (HotNets)*, San Diego, CA, Nov. 2004.

[6] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet routing convergence," *SIGCOMM Computer Communication Review*, vol. 30, pp. 175–187, Aug. 2000.

[7] F. Wang, L. Gao, J. Wang, and J. Qiu, "On understanding of transient interdomain routing failures," in *Proceedings of the 13th IEEE International Conference on Network Protocols*, 2005, pp. 30–39.

[8] Y. Wang, I. Avramopoulos, and J. Rexford, "Design for configurability: rethinking interdomain routing policies from the ground up," *IEEE Journal on Selected Areas in Communications*, vol. 27, pp. 336–348, Apr. 2009.

[9] L. Wang, J. Wu, and K. Xu, "Utilizing route correlation to improve BGP routing convergence," in *9th International Conference on Telecommunications, 2007 (ConTel 2007)*, Jun. 2007, pp. 211–218.

[10] R. Clayton, "Internet multi-homing problems: Explanations from economics," www.cl.cam.ac.uk/~rnc1/shim6.pdf.

[11] D. Clark, B. Lehr, S. Bauer, P. Faratin, R. Sami, and J. Wroclawski, "Overlay networks and future of the Internet," in *Journal of Communications and Strategies*, 2006, pp. 1–21.

[12] D. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, "The case for resilient overlay networks," in *the 8th Annual Workshop on Hot Topics in Operating Systems*, 2001, pp. 152–157.

[13] "Planetlab project," http://www.planet-lab.org/.

[14] L. Tang, Y. Huai, J. Zhou, H. Yin, Z. Chen, and L. Jun, "A measurement study on the benefits of open routers for overlay routing," *Journal of Communications*, vol. 4, pp. 714–723, Oct. 2009.

[15] N. Chowdhury and R. Boutaba, "A survey of network virtualization," *Computer Networks*, vol. 54, pp. 862–876, Apr. 2010.

[16] J. Shamsi and M. Brockmeyer, "Efficient and dependable overlay networks," in *IEEE International Symposium on Parallel and Distributed Processing 2008*, Apr. 2008, pp. 1–8.

[17] Z. Li, L. Yuan, P. Mohapatra, and C.-N. Chuah, "On the analysis of overlay failure detection and recovery," *Computer Networks*, vol. 51, pp. 3828–3843, Sept. 2007.

[18] Y. Zhu, "Routing, resource allocation and network design for overlay networks," Ph.D. dissertation, Georgia Institute Technology, 2006.

[19] J. Shamsi and M. Brockmeyer, "QoSMap: achieving quality and resilience through overlay construction," in *Proceedings of the 2009 Fourth International Conference on Internet and Web Applications and Services*, 2009, pp. 58–67.

[20] Z. Li and P. Mohapatra, "QRON: QoS-aware routing in overlay networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 29–40, Jan. 2004.

[21] D. Adami, C. Callegari, S. Giordano, M. Pagano, and T. Pepe, "Topology design for service overlay networks with economic and QoS constraints," in *Proceedings of the 8th International IFIP-TC 6 Networking Conference*, ser. NETWORKING '09, 2009, pp. 847–858.

[22] H. Zhang, L. Tang, and J. Li, "Impact of overlay routing on end-to-end delay," in *Proceedings of the15th International Conference on Computer Communications and Networks, 2006 (ICCCN 2006)*, Oct. 2006, pp. 435–440.

[23] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," *SIGOPS Operating System Review*, vol. 35, pp. 131–145, Oct. 2001.

[24] G. Rosenbaum and S. Jha, "Resilience provisioning in provider-based overlay networks," in *Proceedings of the The IEEE Conference on Local Computer Networks 30th Anniversary*, ser. LCN '05, 2005, pp. 427–432.

[25] G. Hasegawa, S. Kamei, and M. Murata, "Emergency communication services based on overlay networking technologies," in *Proceedings of the Fourth International Conference on Networking and Services*, 2008, pp. 159–164.

[26] S. Banerjee, C. Kommareddy, K. Kar, B. Bhattacharjee, and S. Khuller, "Construction of an efficient overlay multicast infrastructure for real-time applications," in *INFOCOM 2003*, vol. 2, Mar. 2003, pp. 1521–1531.

[27] Y. Zhu, B. Li, and K. Q. Pu, "Dynamic multicast in overlay networks with linear capacity constraints," *Parallel and Distributed Systems*, vol. 20, no. 7, pp. 925 –939, Jul. 2009.

[28] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Surveys and Tutorials*, vol. 7, no. 2, pp. 72–93, 2005.

[29] M. Fraiwan and G. Manimaran, "Localization of IP links faults using overlay measurements," in *ICC '08*, May 2008, pp. 5629–5633.

[30] J. Kurian and K. Sarac, "Provider provisioned overlay networks and their utility in DoS defense," in *GLOBECOM '07*, Nov. 2007, pp. 474–479.

[31] H. Beitollahi and G. Deconinck, "An overlay protection layer against Denial-of-Service attacks," in *IEEE International Symposium on Parallel and Distributed Processing, 2008 (IPDPS 2008)*, Apr. 2008, pp. 1–8.

[32] H. Haddadi, I. G. Rio, M., A. Moore, and R. Mortier, "Network topologies: inference, modeling, and generation," *IEEE Communications Surveys and Tutorials*, vol. 10, Jul. 2008.

[33] B. Donnet and T. Friedman, "Internet topology discovery: a survey," *IEEE Communications Surveys and Tutorials*, 2007.

[34] M. Malli, C. Barakat, and W. Dabbous, "A survey on Internet topology inference," *Journal of Computer Networks and Internet Research*, vol. 8, pp. 17–30, Jul. 2008. [Online]. Available: http://hal.ccsd.cnrs.fr/docs/00/07/05/68/PDF/RR-5439.pdf

[35] P. Mahadevan, C. Hubble, D. Krioukov, B. Huffaker, and A. Vahdat, "Orbis: rescaling degree correlations to generate annotated internet topologies," *SIGCOMM Computer Communication Review*, vol. 37, pp. 325–336, Aug. 2007.

[36] "the Skitter Project," http://www.caida.org/tools/measurement/skitter/packets/.

[37] "Internet routing registries," http://www.irr.net/.

[38] "GT-ITM topology generator," http://www.cc.gatech.edu/projects/gtitm/.

[39] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "iPlane: an information plane for distributed services," in *Proceedings of the 7th symposium on Operating systems design and implementation*, ser. OSDI 2006, 2006, pp. 367–380.

[40] A. Collins, "The detour framework for packet rerouting," Master's thesis, University of Washington, 1998.

[41] L. Petersen, S. Muir, T. Roscoe, and A. Klingaman, "Planetlab architecture: an overview (ongoing draft)," May 2006.

[42] W. Cui, S. Machiraju, R. H. Katz, and I. Stoica, "SCONE: A tool to estimate shared congestion among Internet paths," Technical Report UCB-CSD-04-1320, Tech. Rep.

[43] S. Roy, H. Pucha, Z. Zhang, Y. C. Hu, and L. Qiu, "On the placement of infrastructure overlay nodes," *IEEE/ACM Transaction on Networking*, vol. 17, pp. 1298–1311, Aug. 2009.

[44] X. Jin, W.-P. Yiu, S.-H. Chan, and Y. Wang, "Network topology inference based on end-to-end measurements," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, pp. 2182–2195, Dec. 2006.

[45] Z. Duan, Z.-L. Zhang, and Y. T. Hou, "Service overlay networks: SLAs, QoS, and bandwidth provisioning," *IEEE/ACM Transaction on Networking*, vol. 11, pp. 870–883, Dec. 2003.

[46] G. Bernstein, D. Cheng, Pendarakis, and D., "Domain to domain routing using GMPLS, OSPF extension v1.1(draft)," Jul. 2002.

[47] "RFC 2328, OSPF Version 2," http://tools.ietf.org/html/rfc2328.

[48] Z. Li and P. Mohapatra, "On investigating overlay service topologies," *Computer*

Networks, vol. 51, pp. 54–68, Jan. 2007.

[49] Z. Li, L. Yuan, and P. Mohapatra, "An efficient overlay link performance monitoring technique," in *Proceedings of Networking 2006*, 2006, pp. 513–524.

[50] J. Kurian, A. Kulkarni, H. T. Vu, and K. Sarac, "ODON: An on-demand security overlay for mission-critical applications," in *Proceedings of the 18th International Conference on Computer Communications and Networks, 2009*, ser. ICCCN '09, 2009, pp. 1–6.

[51] K. P. Gummadi, H. V. Madhyastha, S. D. Gribble, H. M. Levy, and D. Wetherall, "Improving the reliability of Internet paths with one-hop source routing," in *Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation - Volume 6*, 2004, pp. 183–198.

[52] S. Rewaskar and J. Kaur, "Testing the scalability of overlay routing infrastructures," in *PAM 2004*, 2004, pp. 33–42.

[53] M. K. S. D. and B. Umesh, "Availability models for underlay aware overlay networks," in *Proceedings of the second international conference on Distributed event-based systems*, ser. DEBS '08, 2008, pp. 169–180.

[54] K. SD, Madhu and B. Umesh, "A distributed algorithm for underlay aware and available overlay formation in event broker networks for publish/subscribe systems," in *Proceedings of the 27th International Conference on Distributed Computing Systems Workshops*, 2007.

[55] M. Kurant and P. Thiran, "Survivable routing of mesh topologies in IP-over-WDM networks by recursive graph contraction," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 5, pp. 922–933, Jun. 2007.

[56] K. Thulasiraman, M. Javed, and G. Xue, "Primal meets dual: A generalized theory of logical topology survivability in IP-over-WDM optical networks," in *2010 Second International Conference on Communication Systems and Networks (COM-SNETS)*, Jan. 2010, pp. 1–10.

[57] F. C. Ergin, A. Yayimli, and S. Uyar, "An evolutionary algorithm for survivable virtual topology mapping in optical WDM networks," in *Proceedings of the EvoWorkshops 2009*, 2009, pp. 31–40.

[58] W.-L. Yeow, C. Westphal, and U. Kozat, "Designing and embedding reliable virtual infrastructures," in *Proceedings of the second ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures*, ser. VISA 2010, 2010, pp. 33–40.

[59] D. G. Andersen, "Theoretical approaches to node assignment," 2002. [Online]. Available: http://www.cs.cmu.edu/œdga/papers/andersen-assign.ps

[60] R. C. Teixeira, "Network troubleshooting from end-hosts," http://www-rp.lip6.fr/~teixeira/teixeira-manuscript.pdf, retrieved on 28 Sept. 2010.

[61] E. Salhi, S. Lahoud, and B. Cousin, "Heuristics for joint optimization of monitor location and network anomaly detection," in *In ICC 2011 Communications QoS, Reliability and Modeling Symposium (ICC'11 CQRM)*.

[62] L. Denby, J. Landwehr, C. Mallows, J. Meloche, J. Tuck, B. Xi, G. Michailidis, and V. Nair, "Statistical aspects of the analysis of data networks," *Technometrics*, vol. 49, no. 3, pp. 318–334, 2007.

[63] C. Tang and P. K. McKinley, "Topology-aware overlay path probing," *Computer Communications*, vol. 30, pp. 1994–2009, Jun. 2007.

[64] Y. Chen, D. Bindel, H. H. Song, and R. H. Katz, "Algebra-based scalable overlay network monitoring: algorithms, evaluation, and applications," *IEEE/ACM Transaction on Networking*, vol. 15, pp. 1084–1097, Oct. 2007.

[65] A. Shaikh, M. Goyal, A. Greenberg, R. Rajan, and K. Ramakrishnan, "An OSPF topology server: design and evaluation," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 4, pp. 746–755, May 2002.

[66] "CAIDA Archipelago Project," http://www.caida.org/projects/ark/, retrieved on 28 Sept. 2010.

[67] M. Crovella and B. Krishnamurthy, *Internet Measurement: infrastructure, traffic and applications*, 4th ed. John Wiley & Son Ltd, 2006.

[68] X. Zhang and C. Phillips, "Construction of provider-independent overlay networks with high resilience," in *Internet Technology and Applications, 2010 International Conference on*, Aug. 2010, pp. 1–4.

[69] S. Qazi and T. Moors, "On the impact of routing matrix inconsistencies on sta-

tistical path monitoring in overlay networks," *Computer Networks*, vol. 54, pp. 1554–1572, Jul. 2010.

[70] A. Farrel, *The Internet and its protocols: a comparative approach.* Morgan Kaufmann, 2004.

[71] J. Marzo, E. Calle, C. Scoglio, and T. Anjali, "QoS online routing and MPLS multilevel protection: a survey," *Communications Magazine, IEEE*, vol. 41, no. 10, pp. 126–132, Oct. 2003.

[72] "RSVP procotol," http://tools.ietf.org/html/rfc2205.

[73] E. Cela, *The Quadratic Assignment Problem: Theory and Algorithms.* Kluwer Academic Publishers, 1998.

[74] M. Eliane, M. Nair, O. Paulo, and H. Peter, "A survey for the quadratic assignment problem," *European Journal Of Operational Research*, vol. 176, pp. 657–690, 2007.

[75] R. J. Mondragón C, "Optimal networks, congestion and braess' paradox," in *Proceedings from the 2006 workshop on Interdisciplinary systems approach in performance evaluation and design of computer & communications systems*, 2006.

[76] S. Patil and S. Srinivasa, "Theoretical notes on regular graphs as applied to optimal network design," in *ICDCIT'10*, 2010, pp. 236–242.

[77] "Discussion on the girth property of regular graphs," http://www.mathe2.uni-bayreuth.de/markus/reggraphs.html.

[78] "IP-to-AS mapping tools," http://www.team-cymru.org/Services/ip-to-asn.html.

[79] P. Mahadevan, D. Krioukov, M. Fomenkov, X. Dimitropoulos, k. c. claffy, and A. Vahdat, "The Internet AS-level topology: three data sources and one definitive metric," *SIGCOMM Computer Communication Review*, vol. 36, pp. 17–26, Jan. 2006.

[80] B. Bassiri and S. S. Heydari, "Network survivability in large-scale regional failure scenarios," in *Proceedings of the 2nd Canadian Conference on Computer Science and Software Engineering*, ser. C3S2E '09, 2009, pp. 83–87.

[81] Y. Zhu and M. Ammar, "Algorithms for assigning substrate network resources to virtual network components," in *INFOCOM 2006*, Apr. 2006, pp. 1–12.

[82] N. Chowdhury, M. Rahman, and R. Boutaba, "Virtual network embedding with

coordinated node and link mapping," in *INFOCOM 2009*, Apr. 2009, pp. 783–791.

[83] X. Cheng, S. Su, Z. Zhang, H. Wang, F. Yang, Y. Luo, and J. Wang, "Virtual network embedding through topology-aware node ranking," *SIGCOMM Computer Communication Review*, vol. 41, pp. 38–47, Apr. 2011.

[84] N. F. Butt, N. Chowdhury, and R. Boutaba, "Topology-awareness and reoptimization mechanism for virtual network embedding," in *Networking 2010*, 2010, pp. 27–39.

[85] J. Shamsi and M. Brockmeyer, "QoSMap: QoS aware mapping of virtual networks for resiliency and efficiency," in *IEEE Globecom Workshops, 2007*, Nov. 2007, pp. 1–6.

[86] S. R. Ruepp, "Dynamic protection of optical networks," Ph.D. dissertation, Technical University of Denmark, 2008.

[87] "IBM ILOG CPLEX 12.0 software," http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/.

[88] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *INFOCOM 1996*, 1996, pp. 594–602.

[89] "Looking-glass servers," http://www.bgp4.as/looking-glasses.

[90] "RFC 0792 (ICMP Protocol)," http://www.rfc-editor.org/rfc/rfc792.txt, retrieved on 28 Sept. 2010.

[91] M. Coates, A. Hero III, R. Nowak, and B. Yu, "Internet tomography," *IEEE Signal Processing Magazine*, vol. 19, no. 3, pp. 47–65, May 2002.

[92] D. Achlioptas, A. Clauset, D. Kempe, and C. Moore, "On the bias of traceroute sampling: or, power-law degree distributions in regular graphs," in *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, ser. STOC '05, 2005, pp. 694–703.

[93] A. Lakhina, J. Byers, M. Crovella, and P. Xie, "Sampling biases in IP topology measurements," in *INFOCOM 2003*, vol. 1, Apr. 2003, pp. 332–341.

[94] E. Katz-Bassett, H. V. Madhyastha, V. K. Adhikari, C. Scott, J. Sherry, P. Wesep, T. E. Anderson, and A. Krishnamurthy, "Reverse traceroute," in *NSDI'10*, 2010, pp. 219–234.

[95] M. Gunes and K. Sarac, "Importance of IP alias resolution in sampling Internet topologies," in *IEEE Global Internet Symposium, 2007*, May 2007, pp. 19–24.

[96] V. Jacobson, "traceroute tool," ftp://ftp.ee.lbl.gov/traceroute.tar.gz, Feb. 1989.

[97] Y. Shavitt and E. Shir, "DIMES: let the Internet measure itself," *SIGCOMM Computer Communication Review*, vol. 35, pp. 71–74, Oct. 2005.

[98] "TCP-based traceroute tool," http://michael.toren.net/code/tcptraceroute/.

[99] B. Donnet, P. Raoult, T. Friedman, and M. Crovella, "Deployment of an algorithm for large-scale topology discovery," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, pp. 2210–2220, Dec. 2006.

[100] X. Jin, W. Tu, and S. H. G. Chan, "Scalable and efficient end-to-end network topology inference," *IEEE Transaction on Parallel Distributed System*, vol. 19, pp. 837–850, Jun. 2008.

[101] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira, "Avoiding traceroute anomalies with Paris traceroute," in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, ser. IMC '06, New York, NY, USA, 2006, pp. 153–158.

[102] "Paris Traceroute Project," http://www.paris-traceroute.net/newtraceroute.htm.

[103] M. H. Gunes and K. Sarac, "Analyzing router responsiveness to active measurement probes," in *Proceedings of the 10th International Conference on Passive and Active Network Measurement*, ser. PAM '09, 2009, pp. 23–32.

[104] M. Gunes and K. Sarac, "Resolving anonymous routers in Internet topology measurement studies," in *INFOCOM 2008*, Apr. 2008, pp. 1076–1084.

[105] R. Bonica, D. Gan, D. Tappan, and C. Pignataro, "ICMP extensions for multiprotocol label switching," http://www.ietf.org/rfc/rfc4950.txt, Aug. 2007.

[106] F. Viger, B. Augustin, X. Cuvellier, C. Magnien, M. Latapy, T. Friedman, and R. Teixeira, "Detection, understanding, and prevention of traceroute measurement artifacts," *Computer Networks*, vol. 52, pp. 998–1018, Apr. 2008.

[107] B. Augustin, T. Friedman, and R. Teixeira, "Measuring multipath routing in the Internet," *IEEE/ACM Transactions on Networking*, vol. 19, no. 3, pp. 830–840, Jun. 2011.

[108] D. Veitch, B. Augustin, R. Teixeira, and T. Friedman, "Failure control in multipath route tracing," in *INFOCOM 2009*, Apr. 2009, pp. 1395–1403.

[109] R. Bush, O. Maennel, M. Roughan, and S. Uhlig, "Internet optometry: assessing the broken glasses in Internet reachability," in *Proceedings of the 9th ACM SIG-COMM conference on Internet measurement conference*, ser. IMC '09, 2009, pp. 242–253.

[110] X. Jin, W.-P. Yiu, S.-H. Chan, and Y. Wang, "Network topology inference based on end-to-end measurements," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, pp. 2182–2195, Dec. 2006.

[111] X. Jin, Y. Wang, and S.-H. Chan, "Fast overlay tree based on efficient end-to-end measurements," in *2005 IEEE International Conference on Communications 2005 (ICC 2005)*, vol. 2, May 2005, pp. 1319–1323.

[112] X. Jin, Q. Xia, and S.-H. Chan, "On the investigation of path preference in end-to-end network measurements," in *ICC '08*, May 2008, pp. 2091–2095.

[113] X. Jin, W. Tu, and S.-H. Chan, "Traceroute-based topology inference without network coordinate estimation," in *ICC '08*, May 2008, pp. 1615–1619.

[114] B. Yao, R. Viswanathan, F. Chang, and D. Waddington, "Topology inference in the presence of anonymous routers," in *INFOCOM 2003*, vol. 1, Mar. 2003, pp. 353–363.

[115] H. Acharya and M. Gouda, "The weak network tracing problem," in *International Conference on Distributed Computing and Networking*, vol. 5935, 2010, pp. 184–194.

[116] M. Gunes and K. Sarac, "Resolving IP aliases in building traceroute-based Internet maps," *IEEE/ACM Transactions on Networking*, vol. 17, no. 6, pp. 1738–1751, Dec. 2009.

[117] S. Gara-Jimnez and E. Magaa, "Internet mapping at IP level," http://pam2009.kaist.ac.kr/workshop_paper/yourconf1-final9.pdf, 2009.

[118] J. Sherry, E. Katz-Bassett, M. Pimenova, H. V. Madhyastha, T. Anderson, and A. Krishnamurthy, "Resolving IP aliases with prespecified timestamps," in *Proceedings of the 10th annual conference on Internet measurement*, ser. IMC '10,

2010, pp. 172–178.

[119] P. Mérindol, B. Donnet, O. Bonaventure, and J.-J. Pansiot, "On the impact of layer-2 on node degree distribution," in *Proceedings of the 10th annual conference on Internet measurement*, ser. IMC '10, 2010, pp. 179–191.

[120] "The mrinfo project and tool," http://svnet.u-strasbg.fr/mrinfo/index.html.

[121] B. Huffaker, A. Dhamdhere, M. Fomenkov, and K. Claffy, "Toward topology dualism: improving the accuracy of AS annotations for routers," in *Proceedings of the 11th international conference on Passive and active measurement*, ser. PAM'10, 2010, pp. 101–110.

[122] M. Coates, R. Castro, R. Nowak, M. Gadhiok, R. King, and Y. Tsang, "Maximum likelihood network topology identification from edge-based unicast measurements," *SIGMETRICS Performance Evaluation Review*, vol. 30, pp. 11–20, Jun. 2002.

[123] Y. Vardi, "Network tomography: Estimating source-destination traffic intensities from link data," *Journal of the American Statistical Association*, vol. 91, pp. 365–377, Mar. 1996.

[124] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, "Network tomography: recent developments," *Statistical Science*, vol. 19, pp. 499–517, 2004.

[125] N. Duffield, "Network tomography of binary network performance characteristics," *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5373–5388, Dec. 2006.

[126] Y. Gu, G. Jiang, V. Singh, and Y. Zhang, "Optimal probing for unicast network delay tomography," in *INFOCOM 2010*, Mar. 2010, pp. 1–9.

[127] B. Eriksson, G. Dasarathy, P. Barford, and R. Nowak, "Toward the practical use of network tomography for Internet topology discovery," in *INFOCOM 2010*, Mar. 2010, pp. 1–9.

[128] E. Lawrence, G. Michailidis, V. N. Nair, and B. Xi, "Network tomography: A review and recent developments," in *Frontiers in Statistics*. College Press, 2006, pp. 345–364.

[129] P. Sattari, A. Markopoulou, and C. Fragouli, "Multiple source multiple destination topology inference using network coding," in *Workshop on Network Coding,*

Theory, and Applications, 2009 (NetCod '09), 2009, pp. 36–41.

[130] K. Harfoush, A. Bestavros, and J. Byers, "Robust identification of shared losses using end-to-end unicast probes," in *International Conference on Network Protocols, 2000*, 2000, pp. 22–33.

[131] D. Katabi and C. Blake, "Inferring congestion sharing and path characteristics from packet interarrival times," Technical report, MIT LCS, Tech. Rep.

[132] J. Ni, H. Xie, S. Tatikonda, and Y. Yang, "Efficient and dynamic routing topology inference from end-to-end measurements," *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, pp. 123–135, Feb. 2010.

[133] M. Rabbat, M. Coates, and R. Nowak, "Multiple-source Internet tomography," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, pp. 2221–2234, Dec. 2006.

[134] N. Duffield, J. Horowitz, and F. Lo Prestis, "Adaptive multicast topology inference," in *INFOCOM 2001*, vol. 3, 2001, pp. 1636–1645.

[135] N. Duffield, J. Horowitz, F. Lo Presti, and D. Towsley, "Multicast topology inference from measured end-to-end loss," *IEEE Transactions on Information Theory*, vol. 48, no. 1, pp. 26–45, Jan. 2002.

[136] S. Ratnasamy and S. McCanne, "Inference of multicast routing trees and bottleneck bandwidths using end-to-end measurements," in *INFOCOM '99*, vol. 1, Mar. 1999, pp. 353–360.

[137] N. Duffield, J. Horowitz, F. L. Presti, and D. Towsley, "Multicast topology inference from end-to-end measurements," in *ITC Seminar on IP Traffic, Measurement and Modeling*, 2000.

[138] R. Castro, M. Coates, and R. Nowak, "Likelihood based hierarchical clustering," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2308–2321, Aug. 2004.

[139] M.-f. Shih and A. O. Hero, "Topology discovery on unicast networks: A hierarchical approach based on end-to-end measurements," CSPL Technical Report TR-357, Dept. of EECS, Univ. of Michigan, Tech. Rep.

[140] M.-F. Shih and A. Hero, "Hierarchical inference of unicast network topologies based on end-to-end measurements," *IEEE Transactions on Signal Processing*, vol. 55,

no. 5, pp. 1708–1718, May 2007.

[141] J. Ni and S. Tatikonda, "A markov random field approach to multicast-based network inference problems," in *2006 IEEE International Symposium on Information Theory*, Jul. 2006, pp. 2769–2773.

[142] Y. Tsang, M. Yildiz, P. Barford, and R. Nowak, "Network radar: tomography from round trip time measurements," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, ser. IMC '04, 2004, pp. 175–180.

[143] A. D. Pietro, D. Ficara, S. Giordano, F. Oppedisano, G. Procissi, and F. Vitucci, "Merging spanning trees in tomographic network topology discovery," in *ICC '09*, Jun. 2009, pp. 1–5.

[144] Y. Schwartz, Y. Shavitt, and U. Weinsberg, "On the diversity, stability and symmetry of end-to-end internet routes," in *INFOCOM IEEE Conference on Computer Communications Workshops, 2010*, Mar. 2010, pp. 1–6.

[145] M. Gunes and K. Sarac, "Analytical IP alias resolution," in *IEEE International Conference on Communications, 2006. ICC '06.*, vol. 1, June 2006, pp. 459–464.

[146] L. Liu, "Simplifying large-scale communication networks with weights and cycles," Ph.D. dissertation, Queen Mary, University of London, 2010.

[147] A. Urra, J. Marzo, M. Sbert, and E. Calle, "Estimation of the probability of congestion using monte carlo method in ops networks," in *in proceeding of 10th IEEE Symposium on Computers and Communications, 2005 (ISCC 2005)*, Jun. 2005, pp. 561–566.

[148] B. Suman and P. Kumar, "A survey of simulated annealing as a tool for single and multi-objective optimization," *Journal of the Operational Research Society*, vol. 57, pp. 1143–1160, Oct. 2005.

[149] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: the Art of Scientific Computing*, 2nd ed. Cambridge University Press, 1992.

[150] M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Transaction on Modeling Computer Simulation*, vol. 8, pp. 3–30, Jan. 1998.